

Teemu Koski

ALGORITMIAVUSTEISEN SUUNNITTE- LUN HYÖDYNTÄMINEN BETONIELE- MENTTIRAKENTEISEN PARVEKERUN- GON SUUNNITTELUSSA

Diplomityö
Rakennetun ympäristön tiedekunta
Tarkastaja: Professori Mikko Malaska
Tarkastaja: Assistant Professor Kristo Mela
Helmikuu 2020

TIIVISTELMÄ

TEEMU KOSKI: Algoritmiavusteisen suunnittelun hyödyntäminen
betonielementtirakenteisen parvekerungon suunnittelussa
Diplomityö, 70 sivua
Tampereen yliopisto
Rakennustekniikan diplomi-insinööri tutkinto-ohjelma
Helmikuu 2020

Algoritmiavusteisessa suunnittelussa hyödynnetään tietokoneen laskentakapasiteettia algoritmien avulla. Algoritmi on sarja ohjeita, joilla suoritetaan tietty tehtävä. Tässä tutkimuksessa tutkitaan algoritmiavusteisen suunnittelun soveltuvuutta parvekelaatta-, parvekepieli- ja parvekepilarielementtien suunnittelussa.

Tutkimuksen päätavoitteena on luoda parveke-elementtien algoritmiavusteinen elementtisuunnitteluprosessi. Tutkimus koostuu kirjallisuustutkimuksesta ja case-tutkimuksesta. Kirjallisuustutkimuksessa perehdytään algoritmiavusteisen suunnittelun taustaan ja teoriaan sekä parvekkeiden erityispiirteisiin. Lisäksi tutkitaan parveke-elementtien perinteistä elementtisuunnitteluprosessia ja luodaan parveke-elementtien algoritmiavusteinen suunnitteluprosessi.

Case-tutkimuksessa testattiin luodun suunnitteluprosessin toimivuutta. Case-tutkimuksen kohteena on todellinen ja toteutunut asuinkerrostalo. Tutkimuksessa käytettiin Rhinoceros 3D, Grasshopper ja Tekla Structures -ohjelmistoja. Tutkimuksessa havaittiin joitain ongelmia ohjelmistojen välisessä yhteydessä. Ongelmia ilmeni kuitenkin vähemmän kuin aiemmin tehdyssä tutkimuksessa. Parveke-elementtien mallintaminen onnistui tehokkaasti ja algoritmiavusteinen elementtisuunnitteluprosessi oli toimiva, kunhan kohde on riittävän yksinkertainen.

Avainsanat: elementtisuunnittelu, parveke, algoritmi, algoritmiavusteinen suunnittelu

Tämän julkaisun alkuperäisyys on tarkastettu Turnitin OriginalityCheck –ohjelmalla.

ABSTRACT

TEEMU KOSKI: Utilization of algorithm-aided design in the precast balcony element system
Master of Science Thesis, 70 pages
Tampere University
Master's Degree Programme in Civil Engineering
February 2020

Algorithm-aided design utilizes the computing power of a computer through algorithms. An algorithm is a series of tasks that perform a specific task. This thesis investigates the applicability of algorithm-aided design in the element design of precast balcony slab, wall and column.

Main goal is to create the algorithm-aided element design process for balcony elements. Thesis contains literature research and case research. The literature research will explore the theory of algorithm-aided design and the special features of balconies. Additionally, the literature research will explore traditional balcony element design process and will create algorithm-aided element design process for balcony elements.

The usability of the developed design process was studied using a case research. Design software used in the case study were Rhinoceros 3D, Grasshopper and Tekla Structures. Some problems were detected between the link of softwares during the case study. However there were less problems than in the earlier thesis. The balcony elements were successfully modeled, and the algorithm-aided design process was functional as long as the object is simple enough.

Keywords: element design, balcony, algorithm, algorithm-aided design

The originality of this thesis has been checked using the Turnitin OriginalityCheck service.

ALKUSANAT

Haluan kiittää työni ohjausryhmän jäseniä Ilkka Wirkkalaa, Mikko Tonteria, Sanna Lehtosta ja Valtteri Hiltusta Sweco Rakennetekniikka Oy:stä työn aikana annetuista kommenteista ja ideoista. Haluan kiittää myös työn tarkastajaa professori Mikko Malaskaa hyvistä kommenteista.

Tampereella, 02.02.2020

Teemu Koski

SISÄLLYSLUETTELO

1. JOHDANTO	7
1.1 Tutkimuksen tausta	7
1.2 Tutkimuksen tavoitteet ja rajaukset	7
1.3 Tutkimuksen toteutus	8
2. ALGORITMIAVUSTEISEN SUUNNITTELUN TEORIA	10
2.1 Käsitteet ja suunnittelumenetelmien eroavaisuudet	10
2.2 Algoritmiavusteisen mallin taloudellisuus	11
2.3 Algoritmiavusteisen mallin luominen	13
2.4 Rhinoceros 3D ja Grasshopper	15
3. PARVEKKEIDEIN ERITYISPIIRTEITÄ	17
3.1 Parveketyypit	17
3.2 Rakennesuunnittelu	18
3.2.1 Parvekkeen kannatus	18
3.2.2 Parvekkeen jäykistys ja sidonta runkoon	19
3.3 Elementtisuunnittelu	20
3.3.1 Parvekelaatta	20
3.3.2 Parvekepieli	22
3.3.3 Parvekepilari	24
4. ELEMENTTIPARVEKETORNIN ALGORITMIAVUSTEINEN MALLINNUS	26
4.1 Parveke-elementtien perinteinen elementtisuunnitteluprosessi	26
4.1.1 Alustava suunnittelu	27
4.1.2 Tuotantosuunnittelu	28
4.2 Parveke-elementtien algoritmiavusteinen elementtisuunnitteluprosessi	28
4.2.1 Alustava suunnittelu	29
4.2.2 Tuotantosuunnittelu	30
4.3 Algoritmiavusteisella suunnittelulla tavoiteltava hyöty	30
5. CASE: PARVEKE-ELEMENTTIEN ALGORITMIAVUSTEINEN SUUNNITTELU	31
5.1 Lähtökohdat	31
5.2 Parveke-elementtien algoritmiavusteisen suunnitteluprosessin testaus	31
5.2.1 Case-tutkimuksen kohteen esittely	31
5.2.2 Lähtötietojen kokoaminen	32
5.2.3 Elementtien geometrian luominen	34
5.2.4 Elementtien geometrian muokkaus ja elementtien mallintaminen	40
5.2.5 Elementtien detaljointi	42
5.3 Case-tutkimuksessa luotu algoritmi	46
5.3.1 Algoritmin ryhmittely, jäsentely ja käyttöliittymä	47
5.3.2 Algoritmin lähtöparametrit ja yksinkertaistukset	51
5.3.3 Geometrian kokoaminen	52

5.3.4 Parvekelaatta.....	53
5.3.5 Parvekepieli	54
5.3.6 Parvekepilari.....	55
5.3.7 Detajointi.....	55
6.CASE-TUTKIMUKSEN TULOSTEN ANALYSOINTI	60
6.1 Case-tutkimuksen tulosten vertailu toteutuneeseen kohteeseen	60
6.2 Algoritmiavusteisen suunnittelun ongelmat	63
6.2.1 Algoritmia ja mallintamista edeltävät ongelmat.....	63
6.2.2 Algoritmiin ja mallintamiseen liittyvät ongelmat	64
6.2.3 Grasshopperin ja Teklan välisestä linkistä aiheutuvat ongelmat..	66
6.3 Johtopäätökset.....	67
7.YHTEENVETO.....	69
LÄHTEET	70

LYHENTEET JA MERKINNÄT

CAD	Computer-Aided Design, tietokoneavusteinen suunnittelu
Grasshopper	Rhinon visuaalisen ohjelmoinnin lisäosa
Rhino	Rhinoceros 3D -mallinnusohjelmisto
Tekla	Tekla Structures -tietomallinnusohjelmisto

1. JOHDANTO

1.1 Tutkimuksen tausta

Elementtisuunnittelu on perinteisesti tehty tietokoneavusteisesti kaksiulotteisesti piirtämällä tai kolmiulotteisesti mallintamalla. Nykyään elementtisuunnitelmat tehdään yhä useammin 3D-tietomalleista. Tietomallit on perinteisesti tehty manuaalisesti mallintamalla yksittäisiä elementtejä. Elementtien mallinnuksen jälkeen suoritetaan niiden detailointi ja tuotetaan elementtien valmistuskuvat.

Perinteinen elementtisuunnitteluprosessi sisältää paljon suunnittelijan itse manuaalisesti tekemää työtä. Algoritmiavusteisella suunnittelulla voidaan tätä manuaalista työtä vähentää hyödyntäen tietokoneen laskentakapasiteettia. Tämän diplomityötutkimuksen tavoitteena on selvittää, voidaanko algoritmiavusteisen suunnittelun menetelmillä nopeuttaa nykyistä elementtisuunnitteluprosessia ja minkälaisia muutoksia tämä edellyttäisi nykyiseen suunnitteluprosessiin. Algoritmiavusteisen suunnittelun mahdollisuuksia on selvitetty aikaisemmissa tutkimuksissa.

Algoritmiavusteiseen sekä parametrizeeseen suunnitteluun liittyen on aiemmin tehty useita diplomi- ja opinnäytetyötutkimuksia. Toola (2017) tutki parametrin suunnittelun soveltuvuutta teräsrakenteisen sillan pylonin teräskotelon ja teräsosien mallinnukseen. Erkkilä (2017) tutki algoritmiavusteisen suunnittelun hyödyntämistä betonirakenteiden väliseinäelementtien elementtisuunnittelussa. Lalla (2017) tutki kantavien teräsrakenteiden parametrissa rakennesuunnittelua ja mallintamista. Mäenpää (2018) tutki algoritmiavusteisen suunnittelun hyödyntämistä teräsrakenteisen varastohallin rakennesuunnittelussa. Karjalainen (2018) tutki algoritmiavusteisen rakennesuunnitteluprosessin kehitystä teräsrakenteiselle kattorakenteelle. Ketola (2019) tutki teräsristikon algoritmiavusteista suunnittelua ja optimointia. Vähänen (2019) tutki parametrin suunnittelun hyödyntämistä teräsbetonisten runkorakenteiden luonnossuunnittelussa. Hirvikoski (2019) tutki korkeiden betonirakenteiden rakennusten hankekehitysvaiheen stabiiliteettiallin luontia algoritmiavusteisesti.

1.2 Tutkimuksen tavoitteet ja rajaukset

Edellisessä kappaleessa 1.1 luetelluista kahdeksasta tutkimuksesta viisi koskee teräsrakenteita ja kolme betonirakenteita. Betonirakenteita käsittelevissä tutkimuksissa kaksi tutkimusta koskee rakennesuunnittelua ja yksi elementtisuunnittelua. Algoritmiavusteista

elementtisuunnittelua on siis tutkittu teräsrakenteisiin tai rakennesuunnitteluun verrattuna hyvin vähän.

Tämän diplomityön voidaan ajatella olevan jatkoa Erkkilän (2017) diplomityölle *”Algoritmiavusteisen suunnittelun hyödyntäminen betonielementtirakenteiden suunnittelussa”*. Erkkilä tutki omassa diplomityössään algoritmiavusteisen suunnittelun soveltamista väliseinäelementtien elementtisuunnitteluun. Pääpaino Erkkilän tutkimuksessa oli algoritmiavusteisessa mallintamisessa.

Tutkimuksen päätavoitteena on tutkia algoritmiavusteisen suunnittelun soveltuvuutta parveke-elementtien elementtisuunnittelussa. Tutkimus painottuu algoritmiavusteiseen mallintamiseen, sen hyötyjen, haittojen ja mahdollisuuksien tutkimiseen. Tutkimuksen teollisena tavoitteena on luoda algoritmiavusteinen työkalu parveke-elementtien mallintamiseen.

Tutkimus rajataan koskemaan vain elementtisuunnittelua, koska parvekkeiden rakennesuunnittelu on jo varsin tehokasta nykyisellä suunnitteluprosessilla. Toinen merkittävä raja-
 rajaus on tutkia vain pielillä ja pilareilla kannateltuja parvekkeita. Toisin sanoen, case-tutkimuksessa tutkitaan vain betonisia parvekelaatta-, parvekepieli- ja parvekepilarelementtejä. Betoniset parvekekaide-elementit jätetään tutkimuksen ulkopuolelle. Raja-
 rajaus tehdään, jotta tutkimusalue ei kasva liian suureksi ja tutkimus voidaan suorittaa kohtuullisessa ajassa. Raja-
 rajaus mahdollistaa myös riittävän tarkan algoritmiavusteisen suunnittelun erityispiirteiden tutkimisen.

1.3 Tutkimuksen toteutus

Työ toteutetaan kirjallisuustutkimuksena sekä case-tutkimuksena. Kirjallisuusosuudessa perehdytään algoritmiavusteisen suunnittelun taustaan ja teoriaan. Tämän avulla saadaan tarvittava tieto teoriasta case tutkimuksen toteutusta varten. Lisäksi kirjallisuustutkimuksella tuotetaan tietoa perinteisestä sekä algoritmiavusteisesta suunnitteluprosessista.

Case-tutkimuksen tarkoituksena on testata luotua algoritmiavusteista suunnitteluprosessia. Case-tutkimuksen avulla saadaan välitöntä tietoa algoritmiavusteisen suunnittelun soveltuvuudesta parveke-elementtien suunnitteluun. Lisäksi saadaan tietoa käytettävien ohjelmistojen ja niiden rajapintojen rajoitteista sekä prosessin sujuvuuden edellytyksistä.

Ensimmäisessä kappaleessa käsitellään työn tausta, tavoitteet, rajaukset, menetelmä ja rakenne. Toisessa kappaleessa käsitellään algoritmiavusteisen mallinnuksen taustaa sekä tutkitaan algoritmeja, niiden muodostamista sekä ohjelmistoja. Kolmannessa kap-

paleessa käsitellään parveke-elementtien erityispiirteitä. Neljännessä kappaleessa käsitellään nykyistä elementtisuunnitteluprosessia sekä luodaan algoritmiavusteinen elementtisuunnitteluprosessi.

Viidennessä kappaleessa esitetään case-tutkimuksen toteutus sekä tulokset. Siinä testataan uutta algoritmiavusteista suunnitteluprosessia. Kuudennessa kappaleessa analysoidaan case-tutkimuksen tulokset. Viimeisessä kappaleessa arvioidaan kriittisesti tutkimuksen tulosten saavuttamista ja tehdään johtopäätökset sekä yhteenveto tutkimuksesta.

2. ALGORITMIAVUSTEISEN SUUNNITTELUN TEORIA

Tässä kappaleessa perehdytään algoritmiavusteisen suunnittelun teoriaan sekä algoritmiavusteiseen mallinnukseen. Teoria osuuteen perehdytään aiemmin tehtyihin opinnäytetöihin ja kerätään niistä lyhyt tiivistelmä. Teoria osuudessa käsitellään algoritmin ja parametrisuuden käsitteitä sekä algoritmiavusteisen mallin taloudellisuutta ja luomista. Algoritmiavusteisessa mallinnuksessa perehdytään käytettäviin ohjelmistoihin.

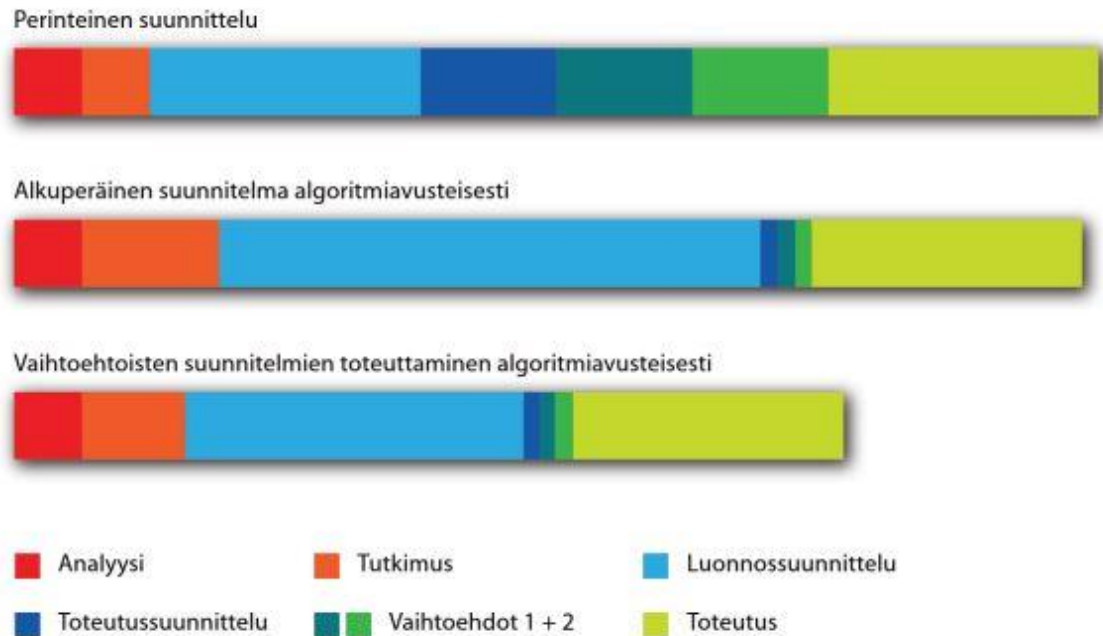
2.1 Käsitteet ja suunnittelumenetelmien eroavaisuudet

Tanska & Österlund (2014) Ketolan (2019) mukaan algoritmilla tarkoitetaan sarjaa ohjeita tai askelia, joilla pyritään saavuttamaan tietty päämäärä. Diplomityössään Ketola (2019) kuvailee algoritmin vaatimuksia. Algoritmin tulee olla kuvattu yksiselitteisesti ja sen jokaisella askeleella on tietty tehtävä. Algoritmin askeleiden järjestys on myös oltava yksiselitteinen. Algoritmilta syötetään lähtötiedoksi tietoalkioita, joita se käsittelee määrätyn logiikan mukaisesti. Suoriuduttuaan algoritmi voi tuottaa dataa jatkojalostettavaksi. Algoritmeja voidaan ketjuttaa ja näin muodostaa pienemmistä algoritmeista suurempia kokonaisuuksia.

Erkkilä (2017) pohtii diplomityössään algoritmiavusteisen suunnittelun ja parametrin mallintamisen käsitteitä. Ne ovat käsitteinä Suomessa melko uusia, eikä alalle ole muodostunut selkeää kuvausta niistä. Tanska & Österlund (2014) Erkkilän (2017) mukaan algoritmiavusteinen suunnittelu ymmärretään ajatustavan muutoksena. Tällä tarkoitetaan uutta tapaa hyödyntää algoritmeja perinteisen suunnittelun kehittämiseksi. Parametrinen mallintaminen taas voi olla algoritmiavusteisen suunnittelun yksi osa. Humppi (2015) Erkkilän (2017) mukaan ehdottaa kokonaan uutta termiä algoritmiavusteinen tietomallintaminen (Algorithm-Aided Building Information Modeling eli AAB). Termi kuvastaa suunnittelutyön tekotapaa sekä suunnittelutyön saavutusta.

Tanska & Österlund (2014) Ketolan (2019) mukaan ovat tutkineet algoritmisen ja perinteisen suunnittelun välisiä eroavaisuuksia. ”Algoritmisessa suunnitteluprosessissa yksittäinen ratkaisu ei ole merkittävä, vaan tehokkuus perustuu nimenomaan lähtötietojen muutoksilla saavutettaviin vaihtoehtoihin ratkaisuihin ja niiden analysointiin.” (Ketola 2019, s. 26), sitaatti kiteyttää lyhyesti algoritmisen suunnittelun hyödyn. Kuvassa 1 on esitetty perinteisen ja algoritmiavusteisen suunnittelun ajankäytölliset eroavaisuudet. Kuvasta huomataan, että algoritmien muodostaminen alkuperäiseen suunnitelmaan ei ole

lopputuloksen kannalta välttämättä nopeampaa kuin suunnittelu perinteisellä menetelmällä. Kuvasta huomataan myös, että algoritmisessa suunnittelussa etu saavutetaan erilaisen suunnitteluratkaisuiden vertailussa, kun algoritmi on jo olemassa.



Kuva 1. Perinteisen ja algoritmiaivusteisen suunnittelun ajankäytölliset eroavaisuudet (Tanska & Österlund 2014, Ketola 2019, s. 26 mukaan).

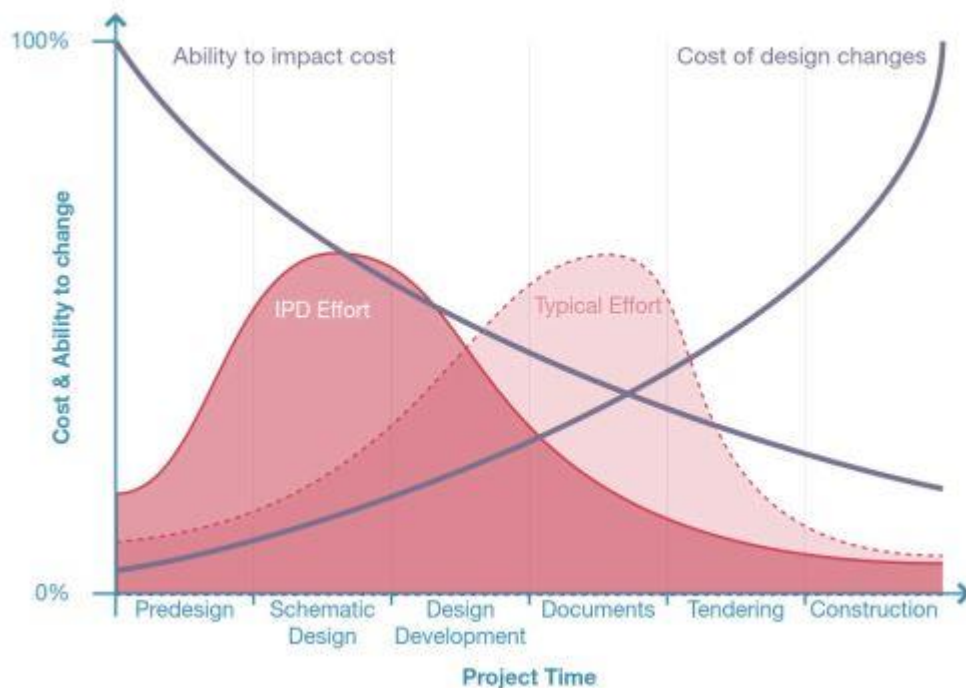
2.2 Algoritmiaivusteisen mallin taloudellisuus

Lalla (2017) esittää diplomityössään neljä tilannetta, joissa algoritmiaivusteisesta (parametrisesta) suunnittelusta on taloudellista hyötyä:

1. geometrialtaan monimutkaiset rakenteet
2. suunnitelmiin on odotettavissa paljon muutoksia
3. kahden tai useamman suunnitteluratkaisun vertailu keskenään ja
4. parametrinen malli on helppo ja nopea luoda. (Lalla 2017, s. 34-37)

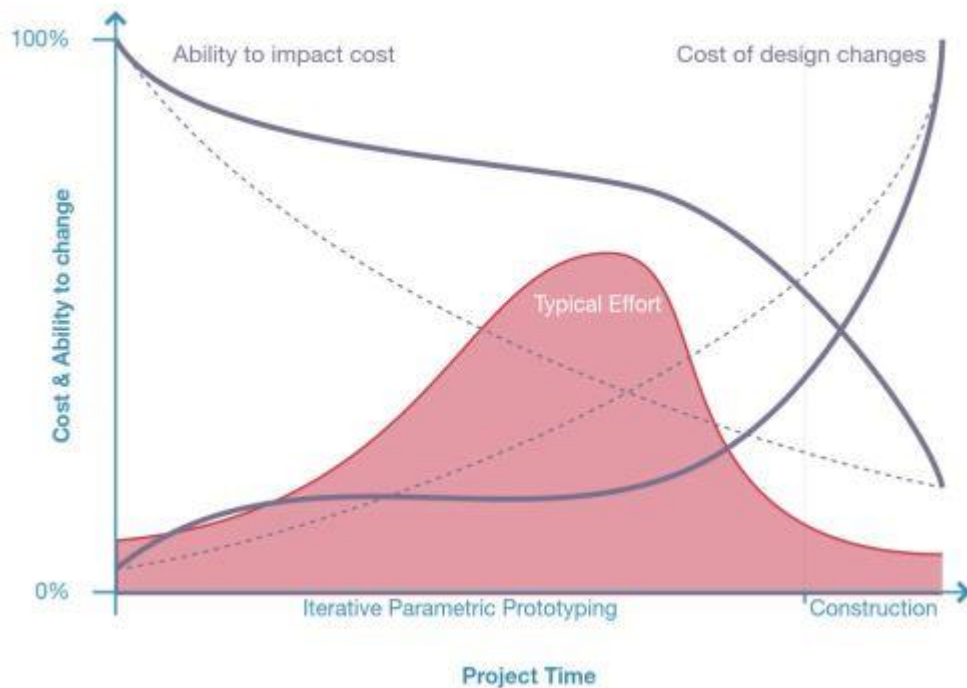
Ensimmäistä tilannetta Lalla havainnollistaa kaksoiskaarevalla kattopinnalla, joka kantellaan kattoristikoiden avulla. Tällaisessa rakenteessa jokainen kattoristikko on erilainen ja jokaisen ristikon geometrian suunnittelu on työlästä tehdä perinteisellä suunnittelumenetelmällä. Algoritmeilla voidaan määrittää tasot, joita ristikoiden paarteet seuraavat, ristikoiden jako sekä uumasauvojen muodostumisen säännöt. Tällä tavalla työmäärä on sama riippumatta kattopinnan muodosta tai ristikoiden määrästä.

Toisen tilanteen Lalla havainnollistaa MacLeamyn kuvaajalla. MacLeamyn kuvaajalla esitetään muutosten ja kustannusten suhdetta projektin etenemiseen. Kuvasta 2 nähdään, että muutosten aiheuttamat kustannukset kasvavat ja kustannuksiin vaikuttaminen pienenee projektin edetessä. Syy tähän on se, että projektin eri osapuolet edistävät omaa suunnittelua ja mitä myöhäisemmässä vaiheessa suunnittelumuutos tehdään, sitä enemmän se vaikuttaa myös muiden osapuolten suunnitelmiin. Davis (2013) Lallan (2017) mukaan esittää, että mikäli projektin suunnittelu toteutetaan parametrisesti, mahdollisuudet vaikuttaa projektin kustannuksiin on olemassa pidempään.



Kuva 2. MacLeamy-kuvaaja (Davis 2013, Erkkilä 2017, s. 54 mukaan).

Kuvasta 3 huomataan, että muutosten aiheuttamat kustannukset pysyvät matalampana pidemmän aikaa verrattuna kuvaan 2. Myös mahdollisuus vaikuttaa kustannuksiin säilyy pidempään.



Kuva 3. Muutettu MacLeamyn kuvaaja (Davis 2013, Lalla 2017, s. 36 mukaan).

Kolmannessa tilanteessa Lalla käsittelee suunnitteluratkaisuiden vertailua. Perinteisellä suunnittelumenetelmällä ratkaisuiden välinen vertailu vaatii, että kaikki ratkaisut on suunniteltu vähintään luonnossuunnitelmien tasolle. Tämä vaatii paljon työtä, jos ratkaisuja on verrattavana suuri määrä. Jos suunnitteluratkaisut parametrisoidaan, voidaan vertailu tehdä helposti parametrejä muuttamalla. Parametrisesti myös suunnitteluratkaisujen optimointi on helpompaa.

Neljännellä tilanteella Lalla tarkoittaa tilannetta, jossa tarvittavat algoritmiset mallit tai mallin osat ovat jo olemassa. Tällöin tilanteeseen sopivan algoritmiavusteisen mallin luominen on helppoa ja nopeaa. Tämä on todennettavissa vain, jos olemassa olevat mallit ovat riittävän hyvin tilanteeseen sopivat, jolloin haluttu suunnittelutulos saavutetaan nopeammin algoritmiavusteisesti kuin perinteisillä menetelmillä.

2.3 Algoritmiavusteisen mallin luominen

Erkkilä (2017) käsittelee diplomityössään laajasti algoritmisen mallin luomista, algoritmien toimivuuden arviointia, algoritmien ryhmittelyä sekä algoritmiavusteisen mallin jäsentelyä ja selkeyttä.

Algoritmiavusteisessa suunnittelussa pyritään siihen, että luotuja algoritmeja voidaan hyödyntää uusissa samankaltaisissa projekteissa. Jotta algoritmeja voidaan hyödyntää myös tulevaisuudessa, vaatii se muutamia uusia taitoja. Woodbury (2010) Erkkilän

(2017) mukaan esittää muutamia uusia taitoja, joilla mahdollistetaan algoritmiavusteinen mallintaminen. Ensimmäinen taito on ymmärrettävän datavirran hallinta. Tämä tarkoittaa, että datavirta, eli komponenttien väliset riippuvuussuhteet, on järjestetty selkeästi siten että komponenttien väliset riippuvuussuhteet ovat ymmärrettävissä (Erkkilä 2017, s. 16-17).

Toinen taito on hajauta ja hallitse -periaate. Siinä ratkaistava ongelma jaetaan pienemmiksi osakokonaisuuksiksi, joista jokainen ratkaistaan erikseen. Lopuksi osat kasataan yhteen kokonaisuuteen, joka ratkaisee alkuperäisen ongelman. Hajauta ja hallitse -periaate johtaa loogiseen modulaarisuuteen, jolloin algoritmin osat ovat päivitettävissä ja korjattavissa yksilöllisesti (Erkkilä 2017, s. 17, Ketola 2019, s. 25).

Kolmas taito on nimeäminen. Selkeällä ja loogisella nimeämiskäytännöllä saadaan tehtyä myös selkeä ja looginen algoritmi. Nimen tulee olla kuvaava ja yksityiskohtainen, jotta siitä on pääteltävissä algoritmin osan toiminta (Erkkilä 2017, s. 17).

Neljäs, viides ja kuudes taito ovat ajattelutapojen muuttamista. Tämä tarkoittaa sitä, että algoritmin luominen ajatellaan prosessina. Ensin mietitään konsepti, jota voidaan hyödyntää tulevaisuudessa uusissa projekteissa. Tämän jälkeen ratkaistava ongelma ajatellaan erillisinä tehtäväsarjoina, jotta kokonaisuuden hahmottaminen ja suunnittelu on toteutettavissa (Erkkilä 2017, s. 18).

Algoritmin toimivuuden arvioinnin tärkeimpinä kriteereinä voidaan pitää muokkaamiseen käytettyä aikaa, taipuisuutta, uudelleen käyttämisen mahdollisuutta ja helppokäyttöisyyttä. Algoritmin tärkeimpiä hyötyjä on sen muokattavuus ja uudelleen käytettävyyys. Muokkaamisella tarkoitetaan parametrusten arvojen, koko algoritmin tai algoritmin osakokonaisuuksien muokkaamista. Muokkaamiseen käytetty aika ei kuitenkaan ole tarkka mittari algoritmin selkeydelle, koska käytetty aika riippuu hyvin paljon muokkaajan taidoista (Erkkilä 2017, s. 19).

Algoritmin taipuisuudella tarkoitetaan sen kykyä sopeutua muutoksiin. Taipuisuus on suoraan yhteydessä algoritmin laajuuteen. Uudelleen käytettävyyys tarkoittaa algoritmin osakokonaisuuksien tai koko algoritmin hyödynnettävyyttä uusissa projekteissa. Uudelleen käytettävyyteen olennaisena osana liittyy algoritmin tarkkuus. Mitä tarkempi algoritmi on, sitä hankalampaa sen uudelleen käytettävyyys myös on (Erkkilä 2017, s. 19).

Helppokäyttöisyydellä tarkoitetaan algoritmisen mallin käyttöä. Sen voidaan ajatella tarkoittavan myös ensimmäistä kertaa mallia käyttävän henkilön kykyä käyttää ja ymmärtää algoritmista mallia.

2.4 Rhinoceros 3D ja Grasshopper

Diplomityössä käytetään McNeel yhtiön ohjelmaa Rhinoceros 3D, David Ruttenin kehittämää lisäosaa Grasshopper sekä Trimble Solutions Corporation kehittämää Grasshopper-Tekla Live Linkiä. Rhino on monimuotoisten viivojen ja pintojen kolmiulotteinen mallinnusohjelma. Viivat ja pinnat perustuvat NURBS-kappaleisiin. NURBS-kappaleet ovat matemaattisesti määriteltäviä, joten ne ovat hyvin tarkkoja verrattuna esimerkiksi mesh-pohjaisiin kappaleisiin. Rhino on helppokäyttöinen ja lisäosien avulla se soveltuu hyvin monimuotoiseen mallintamiseen (Tanska & Österlund 2014, s. 30).

Grasshopper on Rhinoon ilmaiseksi ladattava visuaalisen ohjelmoinnin työkalu. Grasshopperissa on useita erilaisia ohjelmakomponentteja, joita yhdistelemällä muodostetaan algoritmeja. Kaikilla ohjelmakomponenteilla on tietty toiminto, jonka se suorittaa. Grasshopperin toiminta on reaaliaikaista, joten muutokset näkyvät heti. Myös Grasshopperiin on verkosta ladattavissa lisäosia tai lisäosia voi myös itse ohjelmoida perinteisellä tekstimuotoisella ohjelmoinnilla. Ohjelmointikielinä toimivat VB.net, C# ja Python (Tanska & Österlund 2014, s. 30).

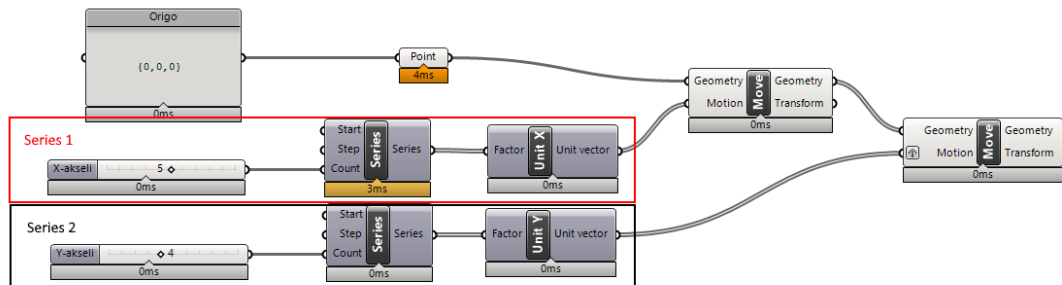
Grasshopper-Tekla Live Link on Grasshopperin ja Tekla Structures -ohjelmistojen välille luotu automaattinen linkki. Linkki sisältää Grasshopper komponentteja, joilla voidaan ohjata Teklaa (Trimble Solutions Corporation). Kuvassa 4 on Grasshopper-Tekla Live Link. Linkki sisältää komponentteja, joilla voidaan ohjata Teklaa. Komponentit on jaoteltu tyyppien mukaan omille välilehdille. Ensimmäinen välilehti Params sisältää komponentteja, joilla voidaan referoida Teklan kappaleita Grasshopperiin. Toisella ja kolmannella välilehdellä on teräs- ja betoniosien mallinnuksen komponentit sekä betonin raudoituskomponentteja. Neljännellä välilehdellä on Teklan kappaleiden muokkauksen komponentteja. Viidennellä välilehdellä on komponentti-, materiaali-, raudoitus- ja profiilikatalogit sekä Teklan kappaleiden asetusten muokkaus komponentit. Viimeisellä välilehdellä on komponentteja, joilla voidaan hakea Teklasta referoitujen kappaleiden tietoja.



Kuva 4. Grasshopper-Tekla Live Link.

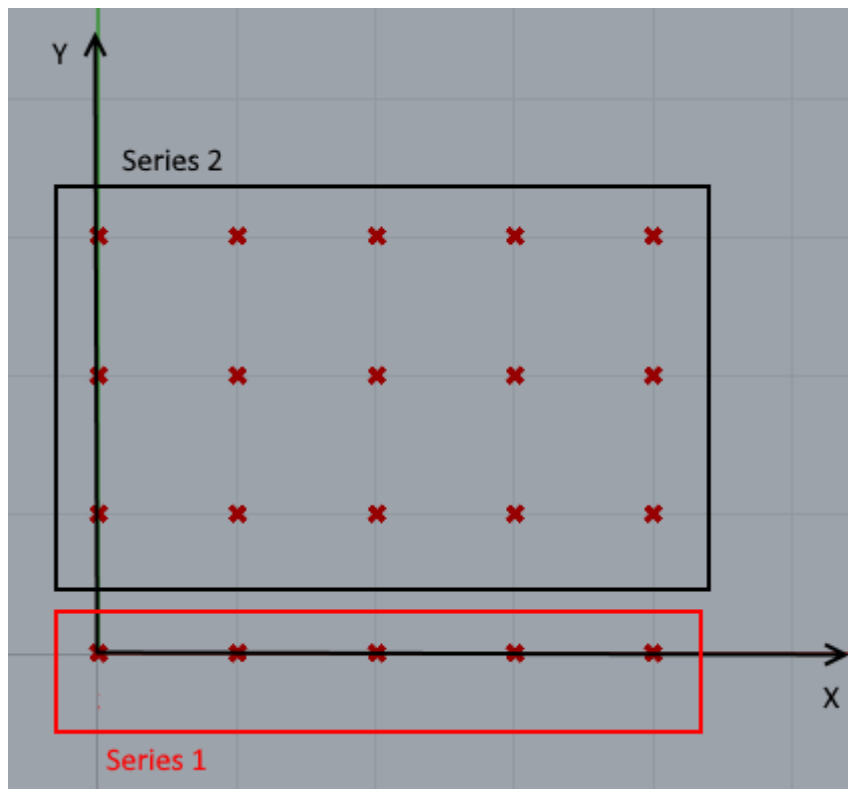
Kuvassa 5 on esitetty esimerkki algoritmista Grasshopperissa. Kuvassa vasemmalla on lähtötiedot: alkupiste, X- ja Y-akselille tulevien pisteiden määrät. Series komponentti muodostaa lukujonon määrättyllä alkuarvolla, askeleella ja askelten lukumäärällä. Tässä tapauksessa alkaen luvusta nolla, askelen ollessa yksi ja lukumäärän ollessa neljä ja viisi. Unit X ja Unit Y muodostavat X- ja Y-suuntaiset yksikkövektorit, joille voidaan antaa

haluttu pituus Factor -kertoimella. Move komponentti siirtää linkitettyä geometriaa linkitetyn vektorin mukaan.



Kuva 5. Esimerkki algoritmista Grasshopperissa.

Kuvassa 6 on esitetty kuvassa 5 esitetyn algoritmin lopputulos Rhinossa. Algoritmi ensin monistaa alkupisteen viisi kerta positiivisen X -akselin suuntaan ja tämän jälkeen monistaa muodostuneet pisteet neljä kertaa positiivisen Y -akselin suuntaan.



Kuva 6. Esimerkkialgoritmin tulos Rhinossa.

3. PARVEKKEIDEN ERITYISPIIRTEITÄ

Tässä osiossa perehdytään parvekkeiden erityispiirteisiin. Ensin käsitellään erilaisia parveketyyppejä. Tämän jälkeen perehdytään parvekkeiden rakennesuunnitteluun niiltä osin kuin se tutkimuksen kannalta on tarpeellista. Lopuksi perehdytään elementtisuunnitteluun elementtityypeittäin.

3.1 Parveketyypit

Parvekejärjestelmät voidaan ryhmitellä parvekkeiden sijoittelun, rakennemallin ja koon mukaan. Rakennemalleja ovat itsekantavat, ulokeparvekkeet sekä ripustetut parvekkeet. Parveke voi olla myös näiden rakennemallien yhdistelmä, eli osittain itsekantava tai osittain rungosta tuettu. Parvekkeet jaotella myös kokonsa puolesta pitkiin parvekkeisiin ja pienparvekkeisiin sekä sijoittelun perusteella rungon ulkopuolisiin ja runkoon sisäänvedettyihin parvekkeisiin (Betoniteollisuus 2010). Tutkimuksessa käsiteltävät parvekkeet ovat esitelty kuvassa 7. Vasemmalla pilarilla ja pielellä kannateltu parveke sekä oikealla pelkillä pielillä kannateltu parveke.



Kuva 7. Pilarilla ja pielellä kannateltu parveke (vasen) ja pelkillä pielillä kannateltu parveke (oikea) (Oma kuva).

Itsekantavat parvekkeet tuetaan perustuksista kantavilla pieliseinillä, pilareilla tai näillä molemmilla. Itsekantavia parvekkeita voidaan käyttää kaikkien runkojärjestelmien ja ulkoseinätyyppien yhteydessä. Ripustettavat parvekkeet voidaan kannatella vetotangoilla kantavasta parvekejulkisivusta, välipohjalaatastosta tai kantavista väliseinistä. Välipohjalaatastosta tai kantavista väliseinistä ripustettaessa julkisivun ei tarvitse olla kantava. Ulokeparvekkeet kannatellaan aina kantavasta välipohjalaatastosta (Betoniteollisuus 2010).

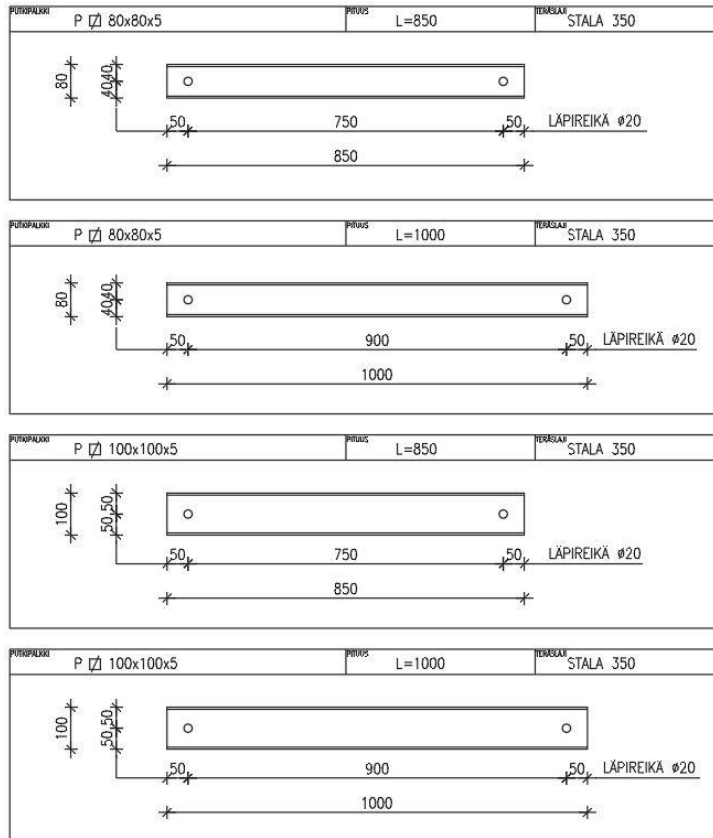
Päällekkäiset parvekkeet ovat yleensä edullisinta ja yksinkertaisinta toteuttaa itsekantavana parveketornina. Tämä edellyttää, että parvekkeiden alapuolalla on tilaa omille perustuksille. Mikäli parvekkeet eivät ala alimmista kerroksista tai ne on sijoitettu epäsymmetrisesti, on järkevä valita joko ripustettu tai ulokeparveke. (Betoniteollisuus 2010)

3.2 Rakennesuunnittelu

3.2.1 Parvekkeen kannatus

Kuten aiemmin todettiin, itsekantavat parvekkeet tuetaan perustuksista kantavilla pieliseinillä, pilareilla tai näillä molemmilla. Parvekkeet sijoitetaan päällekkäin alimman parvekkeen mukaisesti. Kantavat pieliseinät muotoillaan ja mitoitetaan parvekelaatan koon ja päällekkäisten parvekkeiden lukumäärän mukaan. Parvekelaatat tuetaan pääsääntöisesti kerroksittain pieliseinien päältä. Parvekepilarit suunnitellaan yleensä yhden kerroksen korkuisina. Parvekepilarit voidaan suunnitella myös jatkuvina, mutta silloin parvekelaatta kannatellaan pilarissa olevilla konsoleilla. (Betoniteollisuus 2010)

Pieliseinien ja parvekepilarien lisäksi parvekelaatta voidaan kannatella rakennuksen kantavalla rungolla, esimerkiksi kantavilla sandwich-elementtien ulkokuorilla. Parvekelaatan kannatukseen voidaan käyttää myös parvekeputkea. Parvekeputkea käytetään, kun parvekelaatta ei ole kokonaan tuella tai parvekelaatta kannatellaan parvekepilareilla. Parvekeputki on korkealujuusteräksestä valmistettu poikkileikkaukseltaan neliön muotoinen rakenneputki. Kuvassa 8 on esitetty Sweco Rakennetekniikka Oy:ssä käytössä ovat Stala 350 -luokan 80x80x5 ja 100x100x5 rakenneputket. Putkien pituudet ovat 850mm tai 1000mm.



Kuva 8. Parvekeputkien mittapiirustukset (Sweco Rakennetekniikka Oy).

Parvekeputki valetaan parvekelaattaan kiinni elementtitehtaalla. Asennusvaiheessa työmaalla putken vapaa pää asennetaan parvekejulkisivussa olevaan varaukseen ja putkeen kiinnitetään kierretanko sekä mutterit. Lopuksi putki valetaan kiinni rakennuksen runkoon.

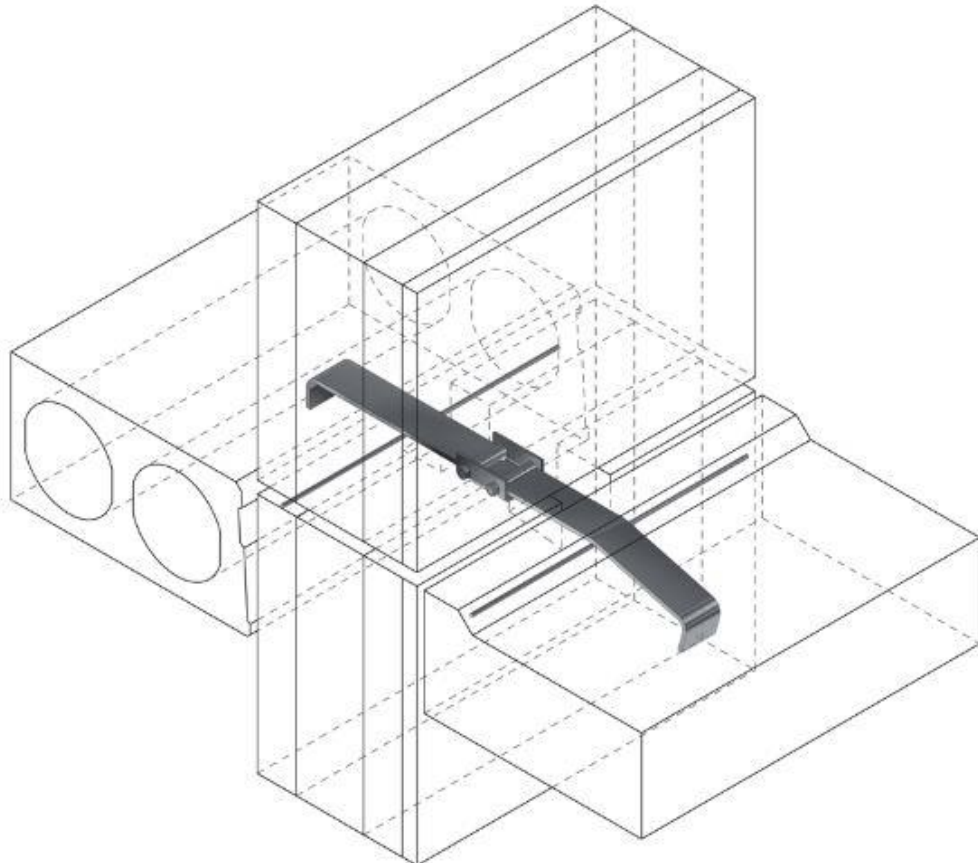
Tässä tutkimuksessa ei käsitellä laattojen ja pieliseinien eikä laattojen ja pilarien välisiä liitoksia. Parvekeputkiliitosta käsitellään vain parvekeputken profiiliin ja pituuden osalta, koska putken mallinnus vaatii kyseiset tiedot.

3.2.2 Parvekkeen jäykistys ja sidonta runkoon

Elementtirakenteisen parveketornin stabiliteetti varmistetaan sitomalla parveketorni kerroksittain rakennuksen runkoon parvekelaatan kohdalta. Sidonta tapahtuu parvekelaatoista ulkoseinän läpi menevillä kiinnitysosilla. Kiinnitysosien valinnassa on huomioitava lämpö- ja kosteusliikkeistä sekä perustusten epätasaisesta painumisesta aiheutuvat liikkeet, vaakavoimat, elementtien valmistus ja siteiden asennustekniset asiat sekä kiinnitysosien kustannukset. Kiinnitysosina käytetään tavallisesti tarkoitukseen kehitettyjä käyttöselosteen omaavia vakioteräsosia. Kaikki kiinnitysosat valmistetaan ruostumattomasta teräksestä, koska ne kulkevat seinärakenteen lämmöneristeen läpi. Kiinnitysosien

sijoittelussa on huomioitava, ettei liitoksiin synny sivuttaisia pakkovoimia. (Betoniteollisuus 2010)

Parveketornin jäykistyksessä yleisesti käytetty kiinnitysosa on parvekesarana. Parvekesarana on vähintään yhdellä nivelellä varustettu teräsosa. Parvekesarana sallii pystysuuntaiset liikkeet ja siirtää siihen kohdistuvia vaakakuormia. (Betoniteollisuus 2010). Kuvassa 9 on esitetty parvekesarana.



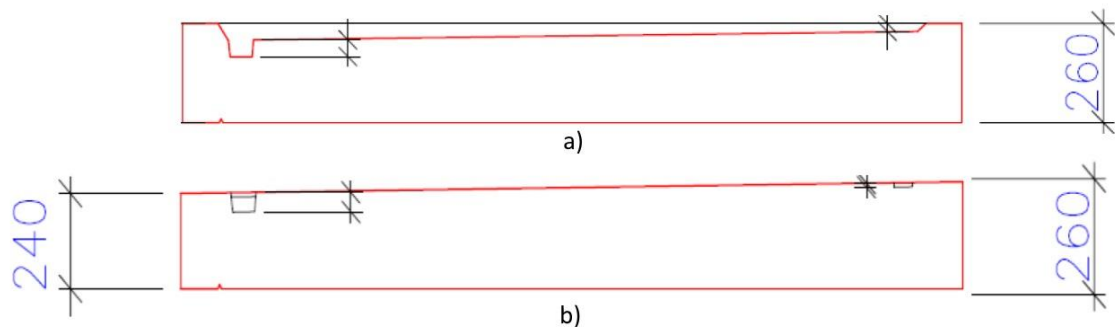
Kuva 9. Esimerkki parvekesaranasta (Peikko Finland Oy 2016).

Parvekesaranan laattaosa valetaan parvekelaattaan elementtitehtaalla. Työmaalla parvekesaranan vapaa pää asennetaan välipohjalaatassa olevaan varaukseen tai ontelolaatan onteloon. Parvekesaranan pultit sekä mutterit kiristetään ja parvekesarana valetaan kiinni kantavaan laattaan (Peikko Finland Oy 2016, s. 5).

3.3 Elementtisuunnittelu

3.3.1 Parvekelaatta

Parvekelaatat jaotellaan vedenpoistojärjestelmän mukaan kiila- ja kuppimallisiin laattoihin. Parvekelaatan yläpinnan kallistus vedenpoistouran suuntaan on 1:80 ja vedenpoistouran kallistus parvekekaivon suuntaan on 1:100. Laatan etu- tai takareunassa sijaitsee vedenpoistojärjestelmän mukainen vedenpoistoura. Tyypillisesti ne laatan reunat, joissa veden valuminen reunan yli on mahdollista, varustetaan tippauralla. Tiipaura on 10x10 mm viiste, joka sijoitetaan laatan alapintaan 100mm laatan reunasta (Betoniteollisuus 2010). Kuvassa 10 on esitetty kiila- ja kuppilaatan poikkileikkaukset mittasuositusten kanssa. Kuvasta havaitaan, että kiilamallisen parvekelaatan etu- ja takareunat ovat eri paksuiset. Laatta kallistaa koko poikkileikkaukseltaan. Muilla reunoilla tyypillisesti kiertää lautaurat. Kuppimallisen parvekelaatan etu- ja takareunat ovat yhtä paksut. Laattaan muotoillaan tasokallistus parvekekaivon puoleista reunaa kohti. Samassa reunassa on myös vedenpoistoura. Kuppilaatan reunoilla kiertää tasainen, laatan paksuinen kaistale.



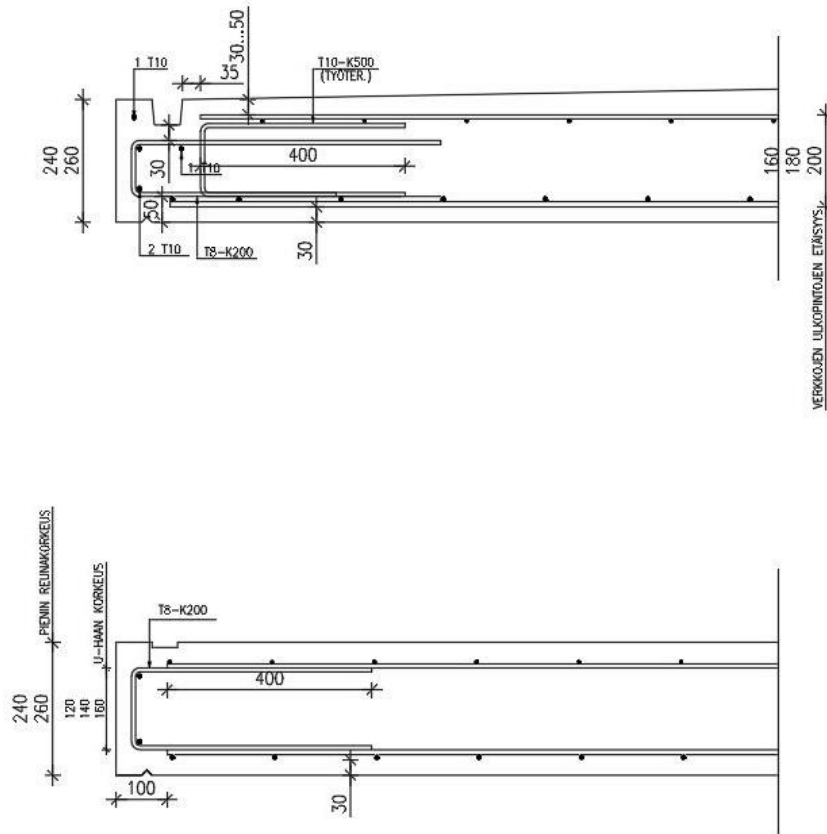
Kuva 10. Parvekelaattojen tyypit: a) kuppimallinen ja b) kiilamallinen. Muokattu lähteestä (Betoniteollisuus 2010, s. 20).

Parvekelaattojen raudoituksessa pyritään työtä säästäviin ja helposti käsiteltäviin ratkaisuihin. Parvekelaatoissa yleisimmät käytettävät teräslaadut ovat verkot B500K ja harjateräkset B500B. Ruostumattomia raudotteita ei pääsääntöisesti ole tarpeen käyttää. Parvekelaattojen raudoituksessa pyritään käyttämään seuraavia perusohjeita (Betoniteollisuus 2010, s. 23).

- yläpinnassa vakioverkkoraudoitus 8-200
- alapinnassa vakioverkkoraudoitus 10-200 tai 10-150, mahdolliset lisäraudoitus tehdään irtotangoilla
- reunahaat T8-k200, reunahaat saman korkuisina tai kahdella eri korkeudella

Kuvassa 11 on esitetty kiilamallisen parvekelaatan päätyjen rauditusperiaate. Kuvasta huomataan, että vedenpoistouran puoleisessa päädyssä reunahaan koko valitaan siten,

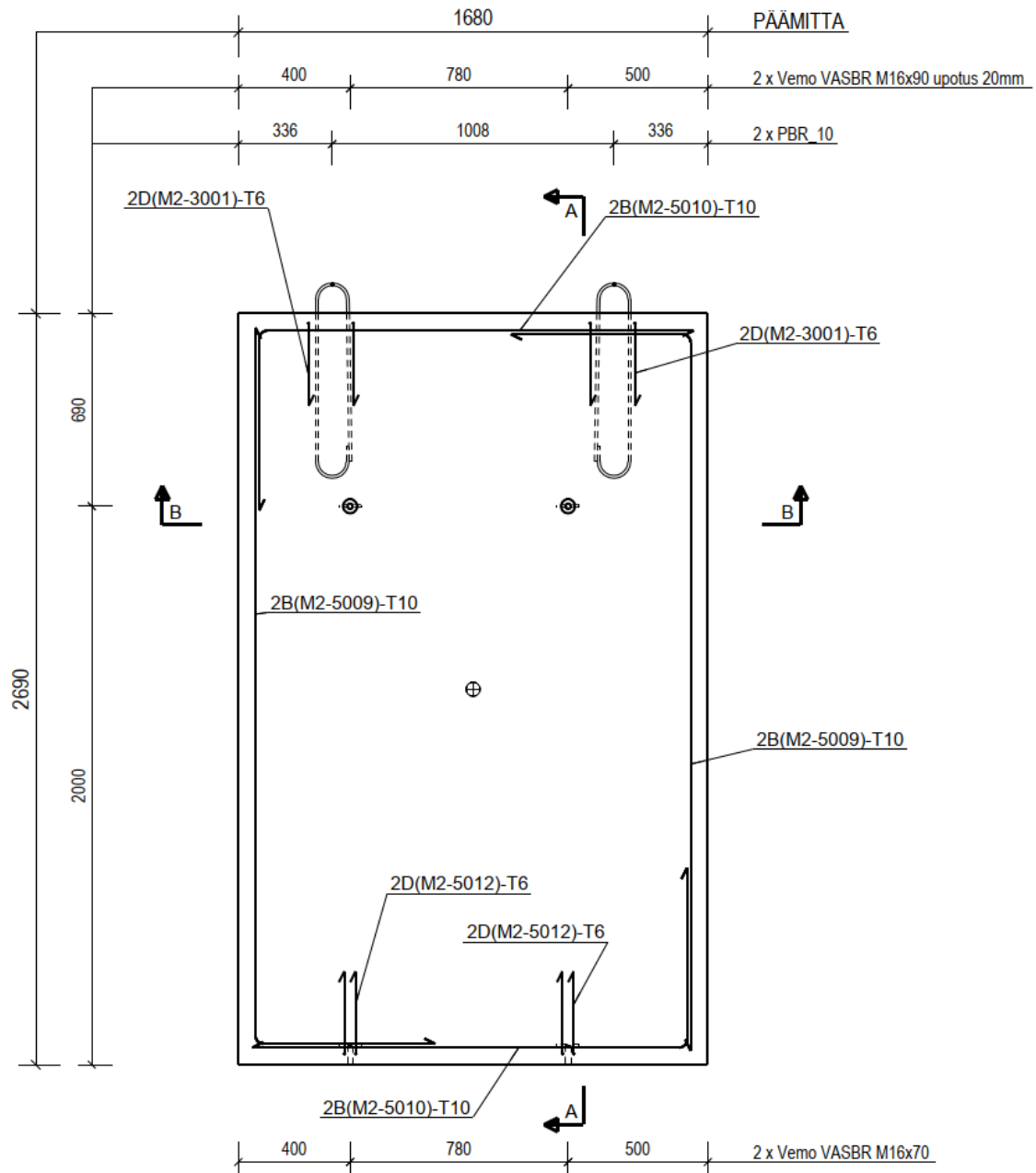
että se mahtuu vedenpoistouran alle. Vedenpoistouran ulkopuoleiseen reunaan sijoitetaan reunateräs vähentämään reunan lohkeamista. Kuppimallisella parvekelaatalla rauditusperiaate on samankaltainen.



Kuva 11. Kiilamallisen parvekelaatan päädyn rauditusperiaate (Sweco Rakennetekniikka Oy).

3.3.2 Parvekepieli

Betonipintaisen pieliementin paksuus on tavallisesti 180 mm. Pielen leveys tulisi olla vähintään 2/3 parvekelaatan leveydestä. Kehäelementin reunojen leveydet tulisi olla vähintään 400 mm. Palkkikannaksen korkeudeksi suositellaan 500 mm, mutta minimi on 300 mm (Betoniteollisuus 2010, s. 21). Kuvassa 12 on esitetty pieliementin valmistuspiirustus.



Kuva 12. Pielielementin valmistuspiirustus.

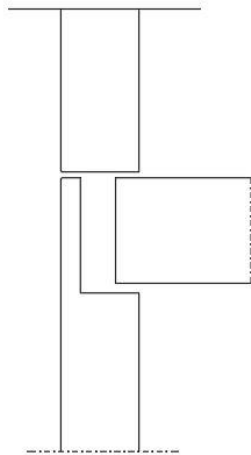
Pielielementti muotoillaan parvekelaatan mukaisesti. Kiilamallisen laatan kanssa pielielementin alareuna viistää laatan kaadon mukaisesti. Kuppimallisen laatan kanssa pielielementin alareuna on tasainen.

Pielielementeissä yleensä käytettävät teräslaadut ovat harjateräs B500B ja mahdolliset verkot B500K. Pielielementeissä ei normaalitapauksissa käytetä verkkoraudotus, ainoastaan ympärikiertävät pieliteräkset 2T10 (Betoniteollisuus 2010, s. 23). Törmäykselle alttiit parvekepielet rakennesuunnittelija tarkastelee ja mitoittaa tapauskohtaisesti.

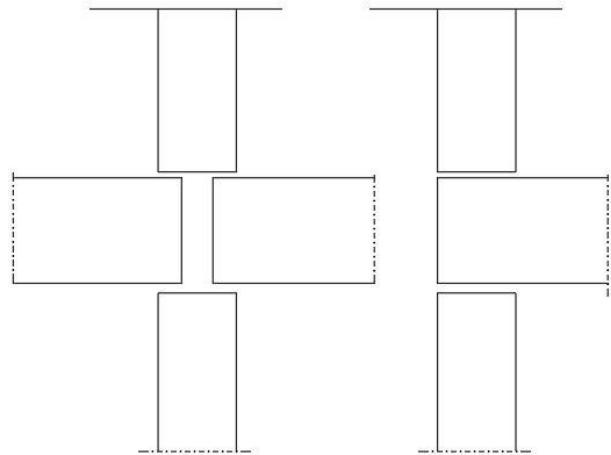
Tässä tutkimuksessa parvekepielet jaetaan reuna- ja keskipieliin. Keskipielellä tarkoitetaan sellaista pieltä, joka jää kokonaisuudessaan parvekelaatan alle. Keskipieleen on

oikeastaan kaksi eri vaihtoehtoa: pieli on pilarimainen tai pielen päälle tulee kaksi parvekelaattaa tuelle. Reunapiellä tarkoitetaan tavallisesti parvekelaatan päädyssä olevaa pieltä. Kuvassa 13 on esitetty reuna- ja keskipielten leikkaukset. Vasemmalla on reunapieli ja oikealla kaksi eri versiota keskipielestä. Kuvasta huomataan, että reunapieli mallinnetaan parvekelaatan yläpinnan tasoon, kun taas keskipieli mallinnetaan parvekelaatan alapuolelle.

REUNAPIELI



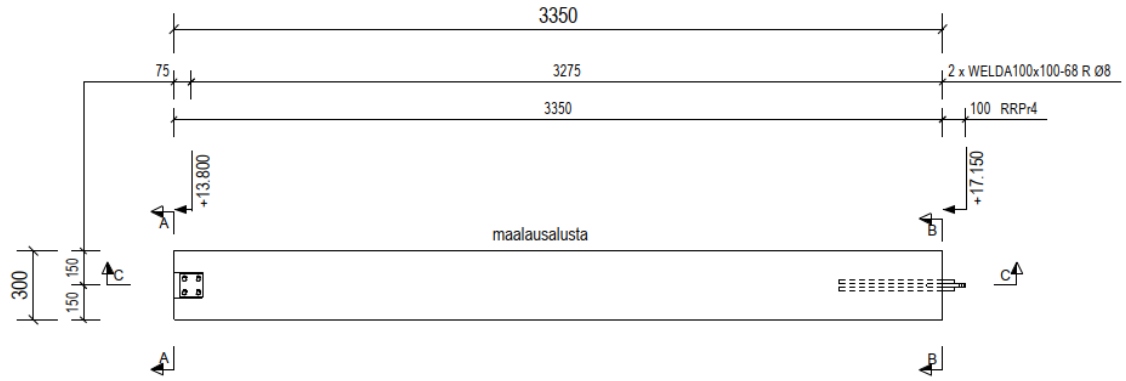
KESKAPIELI



Kuva 13. Reuna- ja keskipielten leikkauskuvat.

3.3.3 Parvekepilari

Parvekepilarit ovat yleensä joko pyöreitä tai suorakaiteen muotoisia pilareita. Pyöreiden pilarien halkaisijan mittasuositus on 230 mm tai 280 mm. Neliöpilareiden sivumitan suositus on 230 mm tai 280 mm. Törmäyskuormalle alttiin pilarin suositus halkaisija tai sivumitta on 380 mm (Betoniteollisuus 2010). Kuvassa 14 on esitetty pilarielementin valmistuspiirustus.



Kuva 14. Pilarelementin valmistuspiirustus.

Pilarelementti valmistetaan joko perinteisesti muotissa valamalla tai ruiskuvaluna. Ruiskuvalussa pyöreän pilarin halkaisija voidaan valita vapaammin kuin perinteisellä valmistusmenetelmällä (Betoniteollisuus 2010). Parvekepilarin raudoitus suunnitellaan aina tapauskohtaisesti.

4. ELEMENTTIPARVEKETORNIN RITMIAVUSTEINEN MALLINNUS

ALGO-

Tässä kappaleessa esitetään parveke-elementtien perinteinen sekä algoritmiavusteinen suunnitteluprosessi. Suunnitteluprosessin automatisoinnissa on hyvin tärkeä tuntee perinteinen suunnitteluprosessi.

4.1 Parveke-elementtien perinteinen elementtisuunnitteluprosessi

Parveke-elementtien perinteinen elementtisuunnitteluprosessin prosessikuvaus on muodostettu tutkijan oman kokemuksen sekä Sweco Rakennetekniikka Oy:n elementtisuunnittelun prosessikaavion perusteella. Prosessikuvaus rajataan alkamaan alustavasta suunnittelusta, jossa mallintaminen alkaa. Prosessikuvaus on listattu alla.

Alustava suunnittelu

1. Raakamallinnus
 - a. elementtijako
 - b. mallinnussuunta/katsomissuunta
 - c. aukkojen ja varausten raakamallinnus
 - d. elementtitunnukset ja numerointisarjat
2. Liitosdetaljit
3. Tyypikohtaiset mallielementit
4. Mallielementtien numerointi
5. Mallielementtipiirustuksen sisäinen tarkastus

Tuotantosunnittelu

1. Elementtien varustelu
 - a. liitokset ympäröiviin rakenteisiin
 - b. valutarvikkeet
 - c. raudoitukset
 - d. numerointi
2. Tuotantopiirustus
3. Suunnittelijan oma tarkastus
4. Toimitus sähkösuunnittelijalle tarvittaessa
5. Sähköjen lisäys tarvittaessa

6. Vastaavan elementtisuunnittelijan (tai RAK-suunnittelijan) tarkastus ja hyväksyntä
7. Kuvan toimitus pankkiin

Ennen alustavaa suunnittelua on suunnittelun valmistelun vaihe, jossa varmistetaan projektin eri osapuolten lähtötietojen yhteensopivuus sekä tehdään projektisuunnitelma, aikataulut, lujuuslaskelmat ja raudoitustiedot.

4.1.1 Alustava suunnittelu

Alustava suunnittelu alkaa raakamallin luonnilla. Raakamalli sisältää käytettävän runkojärjestelmän sekä arkkitehdin määrittämät aukot. Raakamallin tarkkuus voi vaihdella eri tyyppisten kohteiden välillä suurestikin. Asuinrakennuskohteissa raakamallinnuksen teko tapa vaihtelee suunnittelijan tottumuksista riippuen. Esimerkiksi seinät voidaan ensin mallintaa yhtenä levynä ja sen jälkeen pilkkoa elementeiksi tai voidaan mallintaa suoraan elementeiksi. Parveke-elementtien geometria mallinnetaan yleensä suoraan elementtijaon mukaisesti oikein. Elementtiä mallintaessa on huomioitava sen mallinnussuunta. Mallinnussuunta vaikuttaa myös elementin katsomissuuntaan käytettävästä mallinnusohjelmasta riippuen. Elementeille annetaan yhteisesti sovitun käytännön mukaiset elementtitunnukset sekä numerointisarjat.

Elementtien mallinnuksen jälkeen suunnitellaan elementtien väliset liitosdetaljit. Liitossuunnittelu tehdään yleensä CAD-pohjaisesti, koska sen on havaittu olevan helpompaa kuin mallintamalla tehtävä suunnittelu. Liitossuunnittelussa kuitenkin hyödynnetään mallia tarvittaessa.

Liitossuunnittelun jälkeen suunnitellaan jokaisesta erilaisesta elementtityypistä mallielementit. Mallielementiksi pyritään valitsemaan mahdollisimman hankala elementti tai sellainen elementti, jossa on mahdollisimman paljon erilaisia valutarvikkeita. Mallielementit numeroidaan ja niistä luodaan mallipiirustukset. Valitsemalla mahdollisimman vaikea mallielementti, saadaan luotua mahdollisimman kattava piirustus pohja, josta kloonataan loput elementtipiirustukset tuotantosuunnitteluvaiheessa. Mallielementit ja -piirustukset tarkastetaan perinpohjaisesti, jotta turhat virheet saataisiin poistettua riittävän varhaisessa vaiheessa.

Mallipiirustukset on hyvä lähettää tilaajalle sekä elementtitoimittajalle kommentoitavaksi. Varsinkin elementtitoimittajalta voi olla huomioitavia tai muutettavia asioita, joilla he saavat tehostettua omaa toimintaansa. Tässä vaiheessa malliin ja mallipiirustuksiin voidaan tehdä muutoksia ja korjauksia varsin pienellä työllä.

4.1.2 Tuotantosuunnittelu

Alustavan suunnittelun jälkeen alkaa varsinainen elementtien tuotantosuunnittelu. Tuotantosuunnittelu etenee lohkoittain tai kerroksittain riippuen kohteen laajuudesta. Elementit mallinnetaan täysin valmiiksi ennen piirustuksen luontia. Mallinnukseen on syntynyt kokemuksen pohjalta tietty järjestys. Geometrian mallinnuksen jälkeen mallinnetaan liitokset ja liittymät ympäröiviin rakenteisiin. Tässä kohdassa on hyvä yhteensovittaa mallinnusta eri elementtityyppien mallintajien kanssa, jotta liitokset sopivat toisiinsa. On myös syytä huomioida talotekniikan kulkureitit. Esimerkiksi parvekesaranoiden ja -putkien paikkaa miettiessä tulee tarkastaa, etteivät ne törmää lämmityspattereiden putkien kanssa. Liitosten jälkeen mallinnetaan tarvittavat valutarvikkeet ja rauditus sovitulla tarkkuudella. Sovittu tarkkuus tarkoittaa elementtitoimittajan kanssa sovittua tarkkuutta. Lopuksi elementit numeroidaan numerointisarjoittain ja luodaan piirustukset.

Elementtien sisäisten sähkötarvikkeiden merkintään on Sweco Rakennetekniikka Oy:ssä vakiintunut tapa. Siinä elementtipiirustukseen varataan sähkömerkinnöille oma näkymä tai sivu, jossa esitetään elementin geometria sekä valutarvikkeet. Sähkösuunnittelijalle lähetetään elementtipiirustus DWG-muodossa sekä sähkötarvikkeiden merkitsemisohje. Sähkösuunnittelija merkkää sähkötarvikkeet niille varattuun näkymään ja palauttaa merkinnät ennalta sovitulla tavalla. Merkinnät voidaan palauttaa joko DWG-tiedostona tai PDF-tiedostona. DWG-tiedosto sisältää vain sähkömerkinnät, merkintöjen selitykset, kohdistusmerkin sekä sähkönimiön. Elementtisuunnittelija lisää DWG-tiedoston elementtikuvaan ja kohdistaa sen oikealle kohdalle elementtiä. PDF-tiedosto puolestaan sisältää koko sähkösivun kaikkine sisältöineen ja elementtisuunnittelija liittää sen elementtipiirustukseen. Sähkömerkintöjen saavuttua elementtisuunnittelija tarkastaa, että sähkövaraukset mahtuvat elementtiin eivätkä törmäile muiden tarvikkeiden kanssa.

Elementtisuunnittelun lähes joka vaiheeseen liittyy tarkastus. Elementtisuunnittelija tarkastaa aina oman työnsä. Elementtisuunnittelijan oman tarkastuksen jälkeen myös vastaava elementtisuunnittelija tai rakennesuunnittelija tarkastaa ne. Kun suunnitelmat on tarkastettu ja hyväksytty, ne toimitetaan projektipankkiin sovitun käytännön mukaisesti.

4.2 Parveke-elementtien algoritmiavusteinen elementtisuunnitteluprosessi

Tässä kappaleessa tutkitaan parveke-elementtien algoritmiavusteista elementtisuunnitteluprosessia. Prosessi on jaettu neljään omaan osakokonaisuuteen, jotka ovat lähtötietojen kokoaminen, geometrian luominen, geometrian muokkaus ja mallintaminen sekä detaljointi.

4.2.1 Alustava suunnittelu

Prosessi alkaa lähtötietojen kokoamisella. Ensiksi kootaan kaikki tarvittavat lähtötiedot. Lähtötietoina toimivat seuraavat asiat:

1. Arkkitehdin pohjapiirustukset
2. Lujuuslaskelmat
3. Elementtien liitosdetaljit

Arkkitehdin pohjapiirustusten perusteella luodaan parveke-elementtien geometrian lähtötieto. Pohjapiirustuksista poimitaan myös eri kerrosten lattiapintojen korot. Geometrian luomista käsitellään seuraavassa kappaleessa. Lujuuslaskelmat ovat olennaisessa osassa lähtötietojen kokoamisessa. Lujuuslaskennan tuloksena saadaan tarvittavat profiilit, rakenteiden paksuudet sekä materiaalitiedot. Mahdolliset parvekeputket ja niiden sijainnit selviävät myös lujuuslaskennasta. Tässä työssä ei kuitenkaan perehdytä lujuuslaskentaan, vaan oletetaan sen olevan tehty ja tarvittavat tiedot saatu. Elementtien liitosdetaljeista saadaan tarvittavat elementtisaumojen ja pieliseinien valulippojen paksuudet sekä parvekeputken korkoasema.

Lähtötietojen kokoamisen jälkeen muodostetaan elementtien päägeometrialinjat. Tutkimuksen alussa ohjausryhmän kanssa pohdittiin arkkitehdin IFC-mallin hyödyntämistä geometrian lähtötietona. Valittiin muutaman eri kohteen IFC-mallit ja tutkittiin niiden soveltuvuutta geometrian lähtötiedoksi. Kaikissa malleissa havaittiin, että arkkitehdin mallinnustarkkuus ei riitä sovellettavaksi suoraan elementtien mallinnukseen. Näiden IFC-mallien käyttäminen olisi vaatinut IFC-mallin manuaalista muokkausta, kuten Erkkilä (2017, s. 49-50, 60-64) teki diplomityössään. Toinen vaihtoehto on tehdä rakenneosia erottelevia ja muokkaavia algoritmeja.

Edellä kuvatuista syistä johtuen geometrian lähtötieto muodostetaan arkkitehdin pohjapiirustuksen perusteella, joko kaksiulotteisesti piirtämällä CAD-ohjelmistolla DWG-tiedosto tai kolmiulotteisesti mallintamalla Teklassa. Parvekelaatoille määritetään reunamuoto oikean kokoisena sekä pieliseinille ja pilareille määritetään poikkileikkausten reunamuoto.

Kun elementtien päägeometrialinjat on luotu DWG -tiedostoon tai Teklaan, se tuodaan Rhinoon. Tämän jälkeen luodaan algoritmi, joka erittelee laatat, pielet ja pilarit toisistaan. Pielet erotellaan vielä reuna- ja keskipieliin laatan alle jäävän poikkileikkauksen pinta-alan mukaan. Luodaan algoritmi, joka muodostaa elementtien päägeometrialinjat, korkeudet ja monistaa elementit määritettyihin korkoihin. Nämä sisältävät monta erilaista vaihetta ja lopputuloksena on elementtien päägeometriat oikeilla koroilla Rhinossa.

Päägeometrian jälkeen muodostetaan algoritmi, joka mallintaa elementit Teklaan. Algoritmiin määritellään lähtötiedoksi geometrian lisäksi rakenteiden paksuudet ja profiilit sekä nimeämis- ja numerointiasetukset. Algoritmin lopputuloksena on laatta-, pieli- ja pilarielementit mallinnettuna Teklaan sisältäen kaikki tarvittavat tiedot.

4.2.2 Tuotantosunnittelu

Detaljoinnin lähtötietona toimii aikaisemmalla algoritmilla mallinnetut parveke-elementit. Detaljoinnin ensimmäinen vaihe on aukotusten tekeminen. DWG-tiedostoon piirretään tai Teklaan mallinnetaan tarvittavat pieliseinien ja parvekelaattojen aukot. Aukot tuodaan Rhinoon ja niille muodostetaan algoritmit, jotka mallintavat aukot oikeille paikoille. Tämän jälkeen muodostetaan algoritmi, joka rei'ittää pieli- ja laattaelementit.

Aukotusten jälkeen reunapieliin leikataan valulipat. Tämän jälkeen mallinnetaan liittymät muihin rakenteisiin eli parvekeputket ja -saranat, muu detaljointi ja raudoitukset. Kun detaljointi on tehty, tulee kaikki elementit tarkastaa. Jos tarkastuksessa havaitaan virheitä, voidaan korjaukset vielä tässä vaiheessa tehdä kohtuullisella työmäärällä. Kun tarkastus on tehty ja hyväksytty, tehdään elementeistä valmistuskuvat.

4.3 Algoritmiavusteisella suunnittelulla tavoiteltava hyöty

Algoritmiavusteisella suunnittelulla voidaan tavoitella tilanteesta riippuen erilaisia hyötyjä, esimerkiksi ajan ja kustannusten säästö, laadun parantaminen tai vaihtoehtojen tehokkaampi vertailu. Tässä tutkimuksessa luodulla parveke-elementtien algoritmiavusteisella suunnitteluprosessilla pyritään nopeuttamaan sekä tehostamaan raakamallinnusta. Raakamallinnusta tehostamalla syntyy ajan säästön ohella myös kustannusten säästöä. Toinen tavoiteltava hyöty on raakamallin laadun parantaminen. Laadun parantamisella tarkoitetaan mallinnustarkkuuden parantamista sekä inhimillisten virheiden vähentämistä.

5. CASE: PARVEKE-ELEMENTTIEN ALGORITMIAVUSTEINEN SUUNNITTELU

5.1 Lähtökohdat

Case-tutkimuksessa tutkitaan algoritmiavusteisen suunnitteluprosessin toimivuutta parveke-elementtien suunnittelussa. Tutkimuksessa käytettävät ohjelmistot sekä niiden versiot ovat:

- DWG-tiedoston piirtämiseen AutoDesk AutoCAD 2017
- Algoritmin luomiseen Rhinoceros 6 ja Grasshopper December 2019, Build 1.0.0007
- Tietomallinnukseen Tekla Structures 2017i SP7.

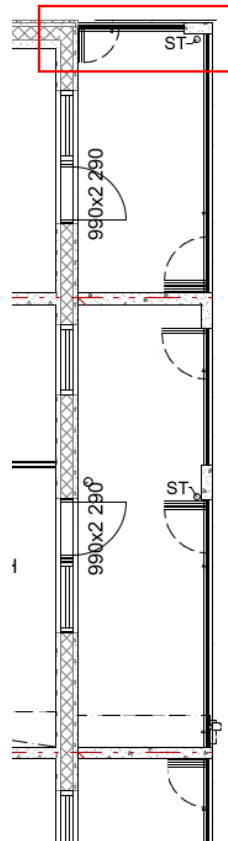
Case-tutkimuksen tavoitteena on kuvata parveke-elementtien algoritmiavusteista suunnitteluprosessia, tutkia algoritmiavusteisen suunnittelun hyötyjä ja haittoja sekä tutkia ohjelmistoihin liittyviä rajoitteita.

5.2 Parveke-elementtien algoritmiavusteisen suunnitteluprosessin testaus

5.2.1 Case-tutkimuksen kohteen esittely

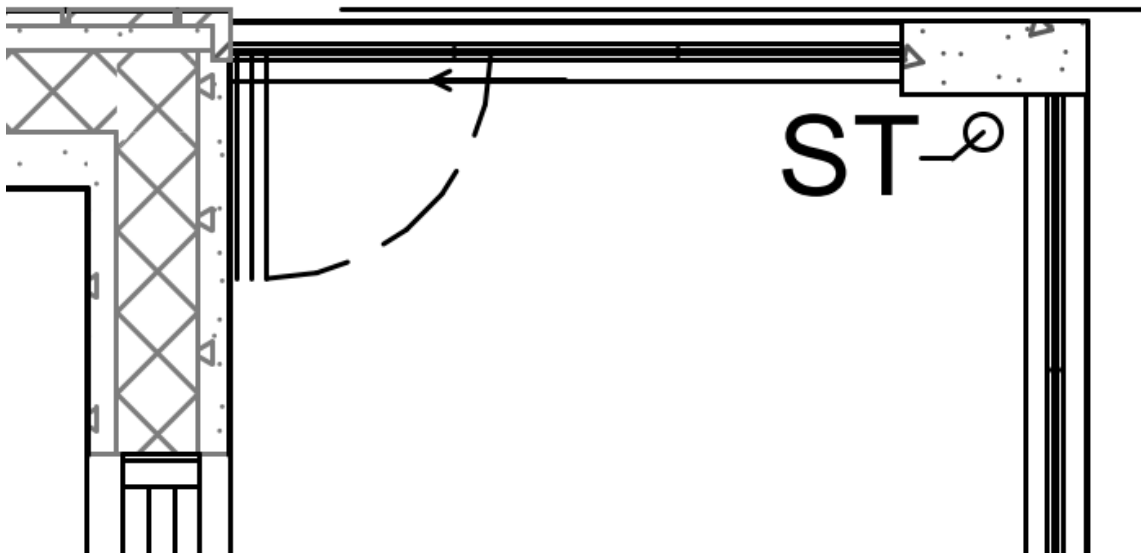
Case-tutkimuksessa on yhden toteutuneen kohteen asuinkerrostalo. Asuinkerrostalossa on neljä erillistä parveketornia rakennuksen eri puolilla. Yksi parveketorni on neljän kerroksen korkuinen ja muut viiden kerroksen korkuisia. Neljä kerrosta korkeassa parveketornissa on rinnakkain neljä parvekelaattaa. Viisi kerrosta korkeissa parveketorneissa yhdessä on rinnakkain kaksi parvekelaattaa ja kahdessa muussa on yksittäiset laatat. Laatat kannatellaan vaihtelevasti pieliseinillä, pilareilla ja parvekeputkillä. Parvekkeet sitotaan rakennuksen runkoon parvekesaranoilla. Kuvassa 15 on Arkkitehtitoimisto AR-Vastamäki Oy:n luoma malli case-tutkimuksessa käytettävästä kohteesta. Kuvassa näkyy kolme neljästä rakennuksen parveketornista.

jos sellainen on kohteesta tehty. IFC-mallin lisäksi tarkastellaan pohjapiirustuksia. Kun kokonaiskäsitys on hahmotettu, varmistetaan, että arkkitehdiltä on saatu uusimmat versiot pohjapiirustuksista. Niistä tarkastetaan, että parvekerakenteet ovat erotettavissa yksiselitteisesti. Kuvissa 17 ja 18 on kuvakaappaukset case-tutkimuksen kohteen toisen kerroksen pohjapiirustuksista. Punaisella suorakulmiolla on merkattu alue, joka esitetään tarkemmin kuvassa 18.



Kuva 17. Kuvakaappaus case-tutkimuksen kohteen toisen kerroksen pohjapiirustuksesta.

Tarkennetusta kuvasta huomataan, että varsinkin parvekelaatan reunat voivat olla vaikeita havaita pohjapiirustuksesta. Arkkitehdit usein mallintavat parvekekaiteet sekä -lasit, joten laatan reunan alueella voi olla useita eri viivoja. Jos laatan oikeaa reunaviivaa ei pystytä pohjapiirustuksesta yksiselitteisesti havaitsemaan, tulee se varmistaa arkkitehdiltä. Kun eri rakenteet on eroteltu yksiselitteisesti, tarkistetaan, että pohjapiirustuksiin on merkattu lattiapintojen korot. Jos korkoja ei ole pohjapiirustuksissa, tulee ne poimia jostain toisesta kuvasta, esimerkiksi rakennuksen leikkauskuvasta.



Kuva 18. Tarkennus kuvan 17 punaisesta suorakulmiosta.

Arkkitehdin kuvien lisäksi varmistetaan, että rakennesuunnittelijan sekä LVIAS-suunnittelijoiden suunnitelmat ovat saatavilla. Rakennesuunnitelmista poimitaan elementtien profiilit, rakennepaksuudet, materiaalitiedot sekä raudoitteet. LVIAS suunnitelmia tarvitaan, jotta vältetään parvekkeiden liitososien ja jäykistävienosien yhteentörmäyksiltä.

5.2.3 Elementtien geometrian luominen

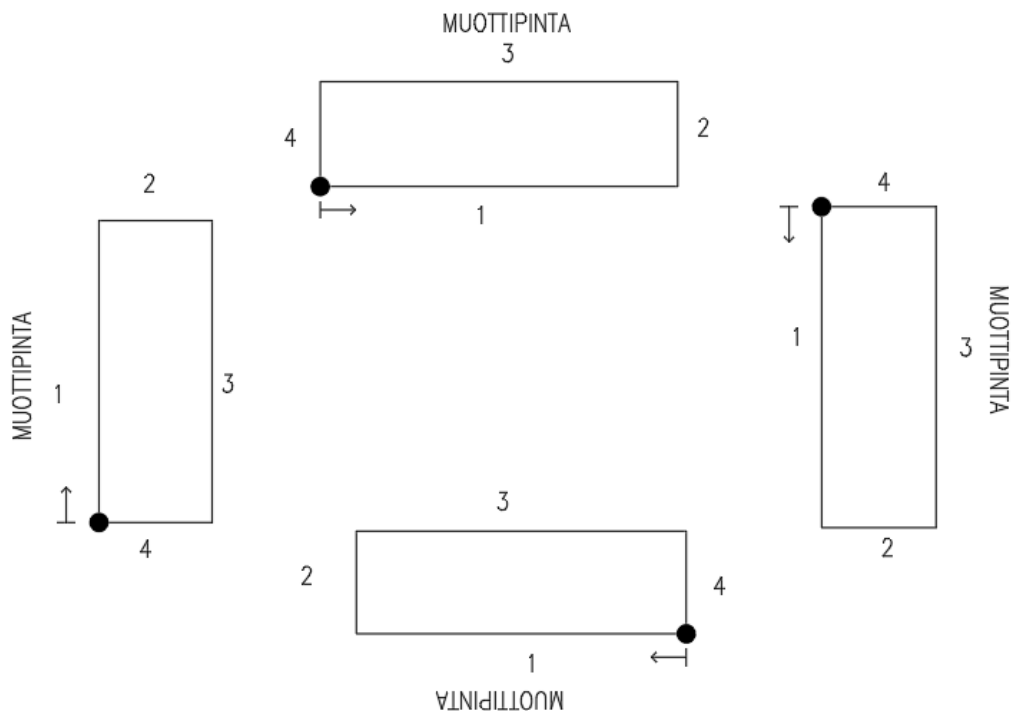
Lähtötietojen kokoamisen yhteydessä tarkastettiin, että eri elementit ovat eroteltavissa yksiselitteisesti. Nyt parveke-elementtien tasogeometria luodaan pohjapiirustuksen perusteella joko piirtämällä tai mallintamalla. Käsitellään ensin piirtämällä luotava tasogeometria.

Pohjapiirustus tuodaan CAD-ohjelmistoon sisään. Pohjapiirustus kannattaa tuoda viitetiedostoksi tai muulla tavalla yhtenä kokonaisuutena. Tällöin pohjapiirustuksen käsittely on helpompaa ja pohjapiirustuksen muokkaamisen riski pienenee. Kun pohjapiirustus on tuotu, luodaan jokaiselle erilaiselle rakenneosalle oma taso eli layer. Layerit nimetään yksilöllisesti, koska algoritmi erottelee rakenneosat layereiden nimien perusteella.

Parvekelaatoille piirretään reunamuoto sekä mahdolliset aukotukset. Reunamuodon piirtämisessä tulee huomioida tarvittava sauma parvekejulkisivuun. Laatan tukipinta pielen päälle tulee piirtää oikein.

Parvekepielille piirretään poikkileikkauksen reunat sekä mahdolliset ikkuna- ja oviaukot. Poikkileikkauksen piirtämisessä tulee huomioida piirtosuuta. Piirtosuunnalla on suora vaikutus algoritmin suorittamaan pielitelementin mallinnuksen suuntaan Teklassa. Mal-

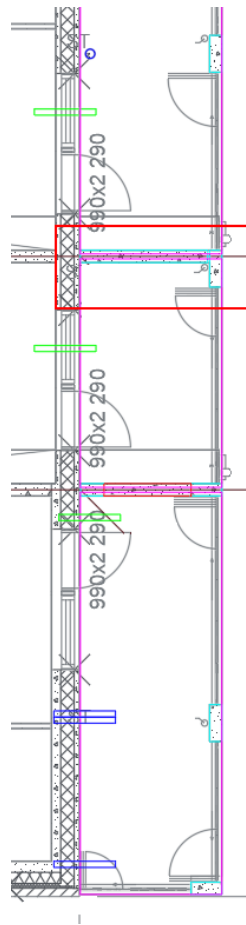
linnuksen suunta taas vaikuttaa elementin katsomissuuntaan elementtikuvassa. Pääsääntö Teklassa seinän mallinnukseen on vasemmalta oikealle, jolloin muottipinta jää seinää edestäpäin katsottaessa seinän takapuolelle. Poikkileikkauksen piirtosuunta noudattaa samaa logiikkaa. Kuvassa 19 on esitetty pielen poikkileikkauksen piirtosuunnan vaikutus elementin muottipintaan. Kaikki suorakulmiot ovat piirretty yksittäisistä viivoista. Numerot tarkoittavat viivojen piirtojärjestystä, musta piste tarkoittaa ensimmäisen viivan aloituspistettä ja nuoli näyttää piirtosuunnan. Kuvasta havaitaan, että kun pielen piirtäminen aloitetaan poikkileikkauksen pidemmästä sivusta, niin mallinnussuunta tulee oikein.



Kuva 19. Parvekepielen piirtosuunnan vaikutus elementin muottipintaan.

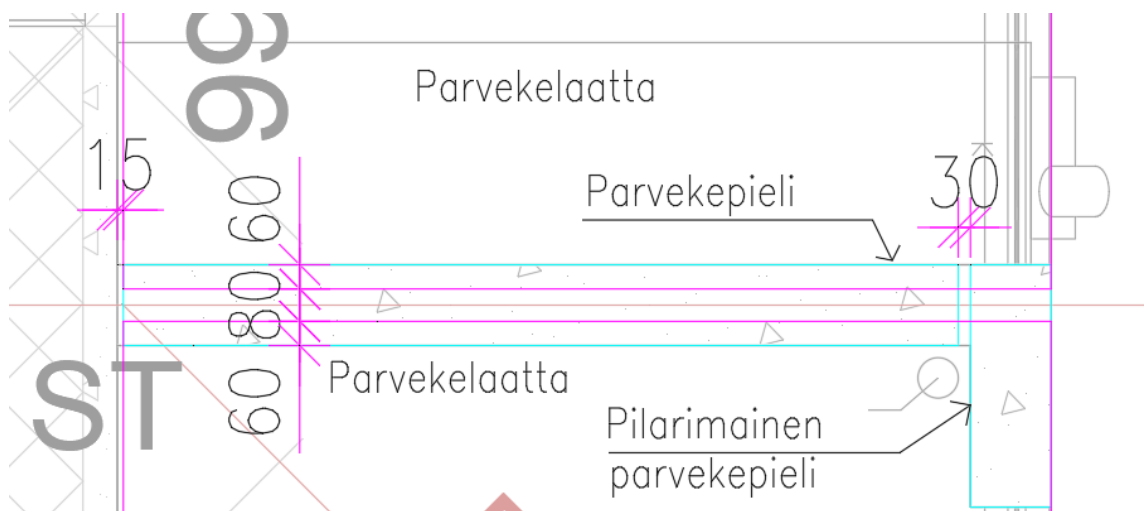
Parvekepilareille piirretään vain poikkileikkaus. Parvekeputkien ja -saranoiden paikat piirretään suorakaiteilla. Kuvassa 20 on AutoCAD:llä pohjapiirustuksen päälle piirretyt rakenneosat. Eri rakenneosat on piirretty eri väreillä, jotta ne erottavat helposti toisistaan. Piirtämisessä käytetyt värit ja rakenneosat ovat:

- Parvekelaatta Violetti
- Parvekepieli Vaalean sininen
- Ovi- tai ikkuna-aukko Punainen
- Parvekeputki Tumman sininen
- Parvekesarana Vihreä



Kuva 20. Pohjapiirustuksen päälle piirretyt rakenneosat.

Kuvassa 21 on tarkennettu kuvan 20 punaisen suorakaiteen osoittamaan kohtaan. Kuvassa on lisätty saumojen ja laatan tukipintojen mitat. Kuvasta nähdään kuinka parvekelaatan ja parvekejulkisivun väliin on jätetty 15 mm sauma, pielen väliin 30 mm sauma sekä laattojen väliin 80 mm sauma. Laattojen tukipinnoiksi on piirretty 60 mm.

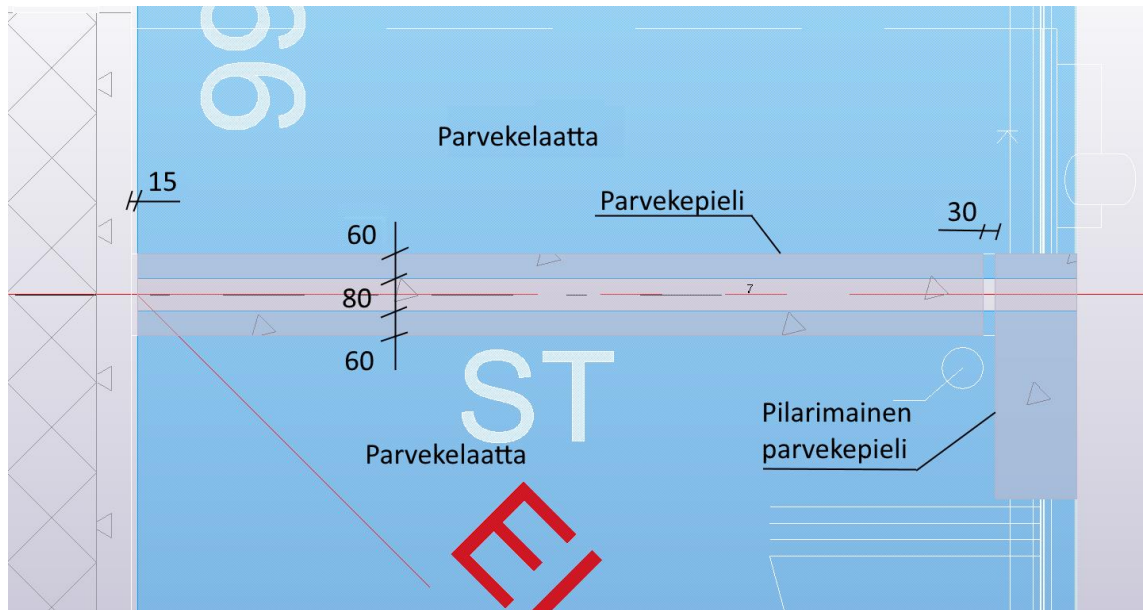


Kuva 21. Parvekelaatan ja -pielen tarkennettu piirustus.

Mallintamalla luotavan tasogeometrian prosessi on pääpiirteittäin samanlainen kuin piirtämällä luotava tasogeometria. Arkkitehdin pohjapiirustus tuodaan Teklaan sisään viitetiedostoksi ja tämän jälkeen mallinnetaan eri rakenneosat käyttäen Teklan Slab, Panel, Column ja Beam -komponentteja. Mallinnetut kappaleet tuodaan Rhinoon käyttäen Grasshopperissa Model object -komponenttia. Kappaleet tulevat Rhinoon kolmiulotteisena. Jotta kappaleita saadaan hyödynnettyä, täytyy luoda algoritmi, joka rikkoo pinnat breb -kappaleiksi ja valitsee niistä z-koordinaatiltaan alimman pinnan. Valittu pinta siirretään vielä XY-koordinaatiston 0-tasolle. Lisäksi muodostetaan algoritmi, joka erottelee päärakenneosat käytetyn Teklan mallinnuskomponentin mukaan. Nimen perusteella erotellaan ikkuna- ja oviaukot, laatan aukot, parvekeputket ja -saranat.

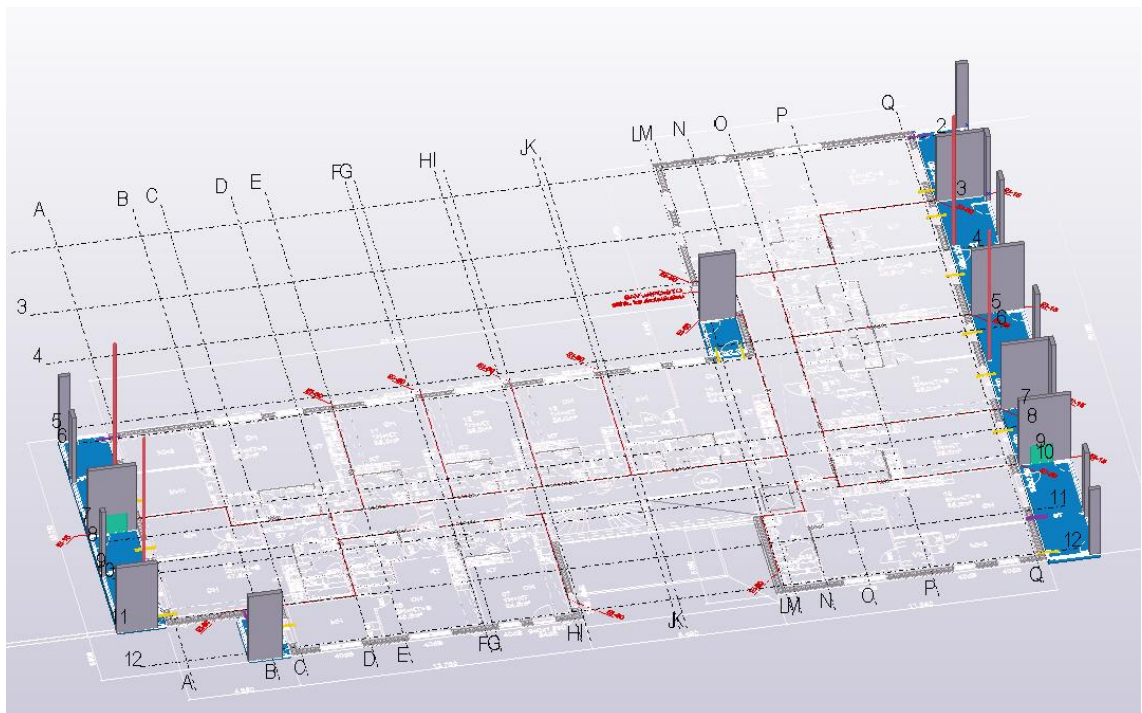
Vastaavasti kuin piirtämällä, parvekelaatalle mallinnetaan reunamuoto sekä mahdolliset aukotukset. Pielille mallinnetaan poikkileikkaus sekä ikkuna- ja oviaukkojen sijainnit. Pielien mallinnussuunnan ja algoritmin mallinnussuunnan välillä on suora yhteys: algoritmin mallinnussuunta noudattaa pielen mallinnussuuntaa täysin. Parvekepilareille mallinnetaan poikkileikkaus. Parvekeputket ja -saranat mallinnetaan suorakaiteen muotoisina palkkeina. Kuvassa 22 on Teklaan mallinnettu rakenneosat pohjapiirustuksen perusteella. Mallinnuksessa käytetyt värit ja rakenneosat ovat:

- Parvekelaatta Sininen
- Parvekepieli Harmaa
- Parvekepilari Punainen
- Ovi- tai ikkuna-aukko Vihreä
- Parvekeputki Violetti
- Parvekesarana Keltainen



Kuva 23. Parvekelaatan ja -pielen tarkennettu mallinnus.

Kuvassa 24 on kuvakaappaus Teklasta, jossa näkyy mallinnetut rakenneosat 3D muo-
dossa. Rakenneosien korkeudella ei ole merkitystä, koska algoritmi laskee rakenteiden
korkeudet määrättyjen lähtöparametrien mukaisesti.



Kuva 24. Kuvakaappaus Tekla mallista.

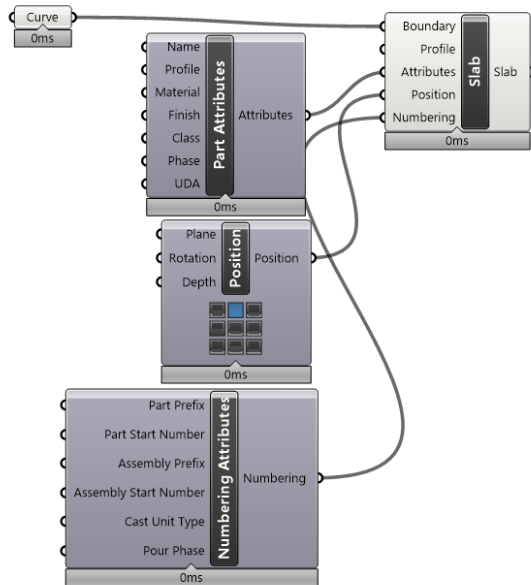
5.2.4 Elementtien geometrian muokkaus ja elementtien mallintaminen

Ennen algoritmin luomisen aloitusta pohdittiin, mitä algoritmin halutaan tekevän. Algoritmin tehtävä kirjattiin sanalliseen muotoon. Tarkoitus on luoda algoritmi, joka poimii ennalta määrätyn tasogeometrian, muokkaa geometriaa tarvittaessa, monistaa geometrian oikeisiin korkoihin ja mallintaa elementit Teklaan. Tämän kirjallisen muodon perusteella tehtävä jaettiin pienempiin osakokonaisuuksiin ja ratkaistiin ne.

Algoritmissa hyödynnetään Trimblen Grasshopper-Tekla Live Link -lisäosaa. Lisäosan toiminta vaatii, että ensin avataan Tekla ja vasta tämän jälkeen avataan Rhino ja Grasshopper.

Ensimmäinen ajatus oli, että ensin muodostetaan kaikki parvekelaatat oikeisiin korkoihin ja mallinnetaan ne Teklaan. Laattojen korkotietojen perusteella lasketaan pielille ja pilareille oikeat korkeudet ja mallinnetaan ne. Tämä ajatus osoittautui toiminnaltaan toimivaksi.

Aluksi kerrosten korkotietojen syöttäminen tehtiin antamalla ensimmäisen kerroksen lattiapinnan korko, kerrosten lukumäärä sekä kerroskorkeus. Tämä kuitenkin havaittiin puutteelliseksi tavaksi syöttää korkotiedot, koska jos rakennuksessa on eri korkuisia kerroksia. Esimerkiksi ensimmäinen tai ylin kerros saattaa usein olla korkeampi kuin muut. Tämä ratkaistiin syöttämällä kaikkien kerrosten lattiapintojen korot erikseen. Parvekekattolaatat muodostetaan samalla geometrialla parvekelaattojen kanssa, mutta niille annetaan oma korko ja ne mallinnetaan erikseen. Parvekekattolaatoilla on erilaiset nimeämis- ja numerointiasetukset kuin parvekelaatoilla. Kuvassa 25 on Grasshopperin Slab -komponentti. Komponentille syötetään ensin geometria, seuraavaksi profiili, eli laatan paksuus, seuraavaksi laatan ominaisuuksia, tämän jälkeen mallinnuspisteiden sijainnin tiedot ja viimeksi numerointiasetukset.



Kuva 25. Slab -komponentti ja sille syötettävät parametrit.

Parvekeputkien ja saranoiden mallinnusta varten parvekelaatoista pitää erotella laatat, joissa on putki ja laatat, joissa ei ole putkea. Erotteluun tehdään algoritmi, joka tunnistaa tukeutuuko laatta pilarin tai pilarimaisen pielen päälle.

Pohjapiirustuksessa on useita viivoja ja usein myös eri rakennekerroksilla on omat varjostukset (eng. hatch). Näistä johtuen tasogeometriaa piirtäessä tai mallintaessa yhtenä ongelmana on käytetyistä ohjelmistoista johtuva pisteen tarttumisen (eng. snap) virhe. Laattojen tukipinnat ovat tärkeässä asemassa rakenteiden kestävyys ja luotettavuuden kannalta. Edellä kuvatuista syistä johtuen laatoille tehtiin vielä algoritmi, joka suorittaa laattojen tukipintojen korjauksen. Kaikkien pielten päälle tuelle tulevien laattojen tukipinnat korjataan kiinteään arvoon.

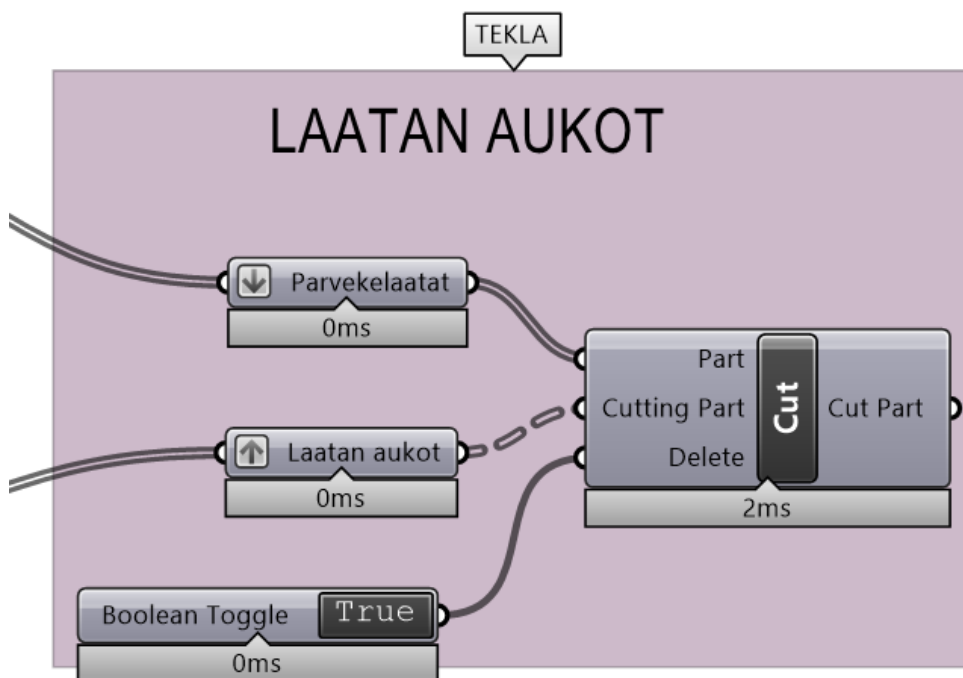
Kun laatat oli saatu mallinnettua, oli vuorossa parvekepielet. Ensin muodostetaan algoritmi, joka hakee parvekepielen poikkileikkauksesta keskilinjan. Seuraavaksi täytyy erottaa keski- ja reunapielet toisistaan. Erotteluun muodostetaan algoritmi, joka tarkistaa montako laattaa tulee pielen päälle tuelle sekä vertaa laatan ja pielen leikkauspinta-alaa pielen poikkileikkauksen pinta-alaan XY-tasossa. Erottelun jälkeen täytyy laskea parvekepielen alapintojen korot sekä pielen korkeudet parvekelaattojen korkotietojen ja elementtien välisten saumojen paksuuksien perusteella. Tällä tavalla saadaan mallinnettua kaikki muut pielet alinta ja ylintä lukuun ottamatta. Alimman pielen korkeus lasketaan anturan yläpinnan, ensimmäisen kerroksen laatan alapinnan sekä saumojen paksuuksien mukaan. Ylimmän pielen korkeus lasketaan ylimmän parvekelaatan yläpinnan, parvekekattolaatan alapinnan sekä saumojen paksuuksien mukaan. Lopuksi mallinnetaan parvekepielet Teklaan.

Parvekepielien jälkeen vuorossa oli parvekepilarit. Parvekepilarien algoritmin periaate on samankaltainen kuin parvekepielillä. Ensin haetaan pilarien alapintojen korot, laskeaan pilarien korkeudet laattojen korkojen ja elementtien välisten sauma-arvojen perusteella. Lopuksi mallinnetaan parvekepilarit Teklaan.

Kaikille Teklaan mallinnettaville elementeille tulee määrittää numerointiasetukset. Teklassa numerointiin vaikuttavat elementtien prefix sekä start number. Numerointiasetukset sovitaan aina etukäteen projektin alussa kaikkien eri osapuolten kesken. Yleensä parveke-elementtien osalta erilaisilla elementeillä on erilaiset kiinteät prefixit, mutta start number saattaa vaihdella eri projektien välillä. Tästä syystä tässä diplomityössä luotu algoritmi määrittää elementeille prefixit, mutta ei start numberia.

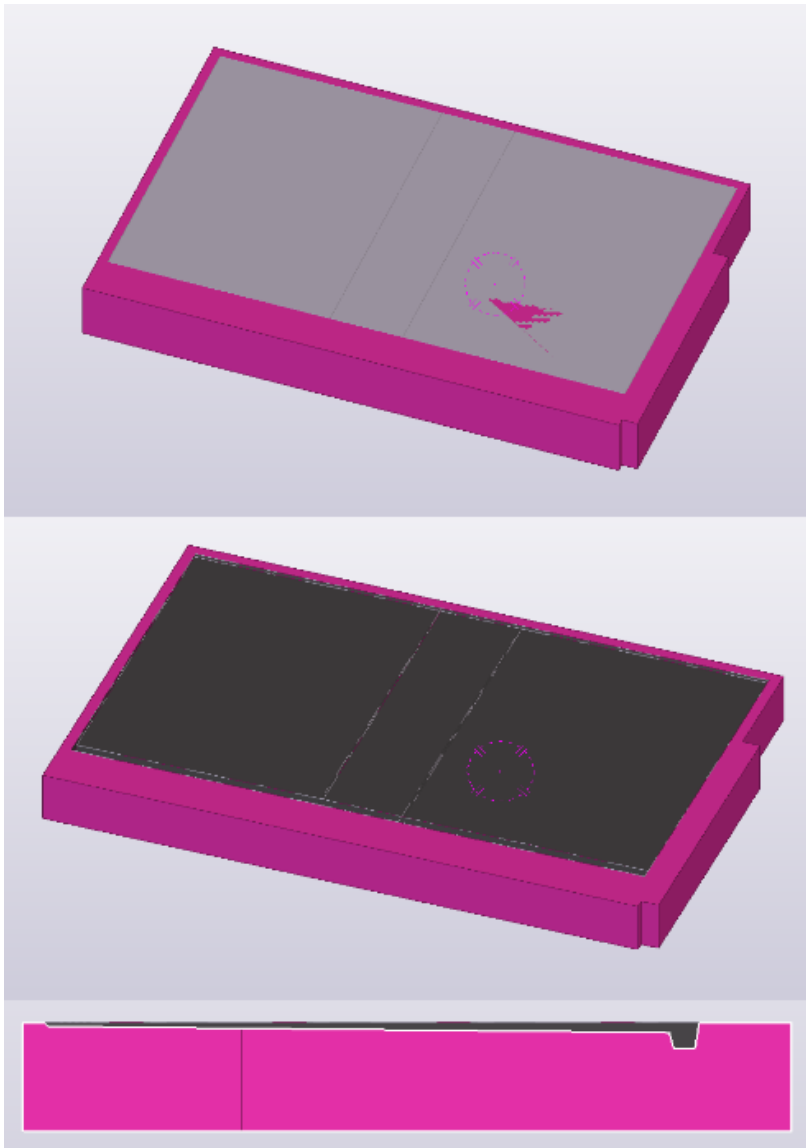
5.2.5 Elementtien detaljointi

Algoritmiavusteisen parveke-elementtien suunnitteluprosessin mukainen detaljoinnin ensimmäinen vaihe on aukotusten tekeminen. Ensin tehdään laattojen aukotukset, jos sellaisia on. Ensin mallinnetaan aukkojen kokoiset kappaleet halutuille koroille. Ohjelmistoteknisistä syistä leikkaavaa kappaletta venytetään ylös- ja alaspäin, jotta leikkaavan ja leikattavan kappaleen pinnat eivät ole tasan. Aukotukset tehdään Part cut -komponentilla. Part cut -komponentille syötetään ensin leikattavat osat, seuraavaksi leikkaavat osat ja lopuksi valitaan, poistetaanko leikkaavat osat mallista leikkauksen jälkeen. Kuvassa 26 on esitetty part cut -komponentti, joka leikkaa laattojen aukot.



Kuva 26. Cut part -komponentti, jolla leikataan parvekelaattojen aukot.

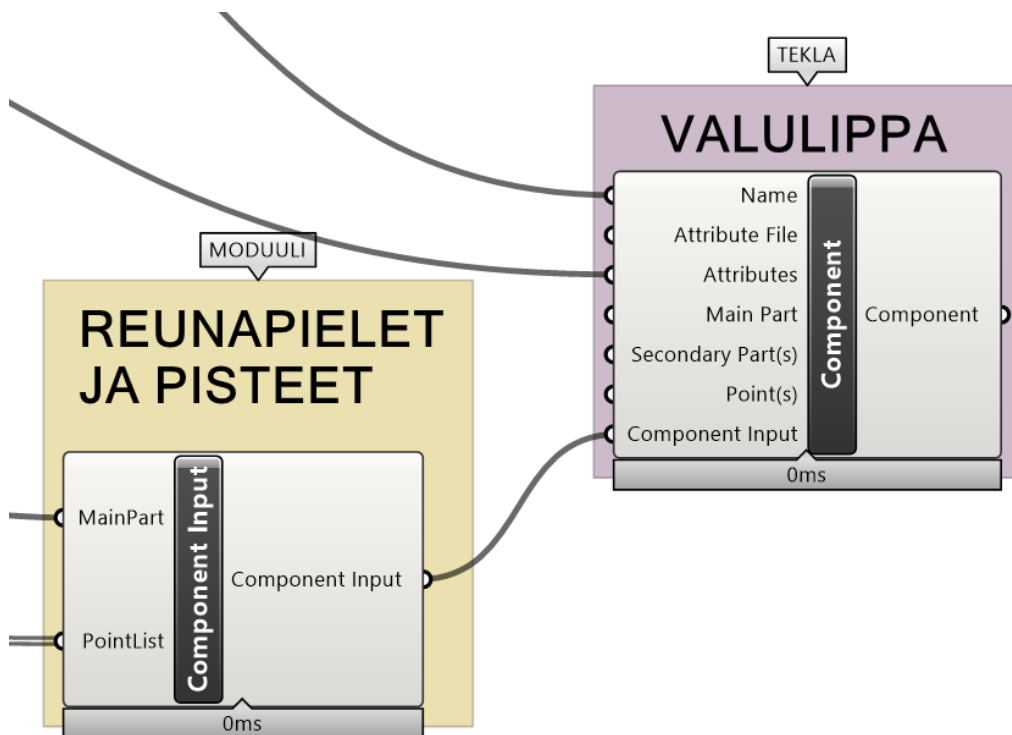
Parvekelaattojen vedenpoistoon liittyvät urat ja kallistukset ovat melko hankalia mallintaa. Teklassa ei ole olemassa pluginia, joka tekisi kallistukset, joten ne on perinteisesti pitänyt mallintaa leikkauskappaleella. Tutkijan oman kokemuksen mukaan tehokas tapa kallistusten ja urien mallintamiseen on mallintaa Teklan Slab -työkalulla kallistuksen muoto laatan molemmista suunnista ja leikata parvekelaattaa muodolla. Kuvassa 27 on esitetty parvekelaattaan perinteisesti mallinnettu vedenpoistoura ja -kallistus. Ylimpänä näkyy Slab -työkalulla mallinnettu uran ja kallistuksen muoto, keskellä muoto on leikattu parvekelaattaan ja alimpana on leikkaus parvekelaatasta, jossa näkyy uran ja kallistuksen muoto. Lisäksi tasokallistuksen pitäisi olla 1:80 ja uran kallistuksen 1:100. Näistä syistä johtuen parvekelaatan vedenpoiston algoritmiavusteiseen mallintamiseen ei saatu ratkaisua tutkimuksen aikana.



Kuva 27. Parvekelaattaan perinteisesti mallinnettu vedenpoistoura ja -kallistus.

Parvekepielten ikkuna- ja oviaukkojen leikkaamista varten tulee mallintaa leikkauskappaleet. Ensin poimitaan lähtötietogeometriasta aukkojen sijainnit. Seuraavaksi muodostetaan aukon keskilinja, jonka mukaan aukko mallinnetaan. Ikkuna-aukoille annetaan kiinteällä parametrilla aukon korkeus ja aukon alareunan korkeus pielen alareunasta. Oviaukon korkeus määräytyy myös kiinteällä parametrilla ja oviaukko leikataan aina pielen alareunasta lähtien muodostaen pielestä kehäelementin.

Aukotusten jälkeen vuorossa on reunapielten valulippojen leikkaus. Valulippojen leikkausta varten keski- ja reunapielet ovat eroteltu toisistaan. Seuraavaksi muodostetaan algoritmi, joka rikkoo reunapielen poikkileikkauksen yksittäisiksi viivoiksi, valitsee viivoista parvekelaatan keskipistettä lähinnä olevan. Tällä tavoin saadaan eroteltua pielestä se puoli, johon leikkaus tehdään. Leikkaus tehdään aina koko pielen matkalle. Itse leikkaus tehdään Sweco Rakennetekniikka Oy:n kehittämällä Teklan pluginillä eli lisäosalla. Teklan plugineita voidaan lisätä Grasshopperilla Component -komponentilla. Siihen määritetään käytettävän komponentin nimi, käytettävät asetukset, pääosa, liittyvä osa sekä pisteet. Plugin vaatii toimiakseen pääosan ja kaksi pistettä. Pääosa ja pisteet on valittava oikeassa järjestyksessä ja oikean järjestyksen luomiseen käytetään Grasshopperissa Component Input -komponenttia. Siihen määritellään syötettävät tiedot oikeaan järjestykseen ylhäältä alaspäin. Kuvassa 28 on esitetty Component Input -komponentti sekä valulipat leikkaava Component -komponentti.

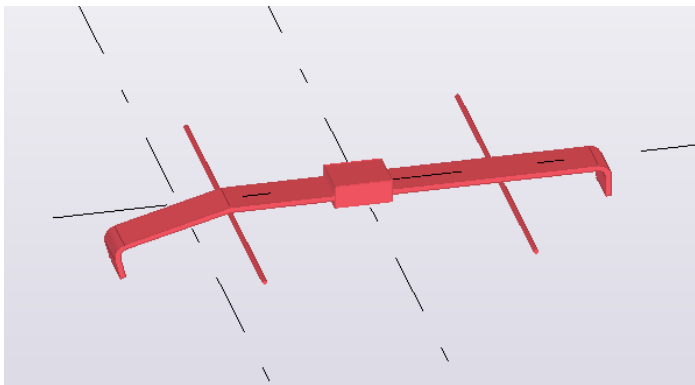


Kuva 28. Reunapielten valulipat leikkaava komponentti.

Valulipan paksuus määritellään kiinteällä parametrilla. Leikkauksen korkeus ja syvyys lasketaan laatan paksuuden, laatan tukipinnan ja elementtien välisten saumojen paksuuksien mukaan.

Valulippojen jälkeen prosessin mukaisesti mallinnetaan liittymät muihin rakenteisiin. Tällä tarkoitetaan parvekelaattojen parvekeputkia- ja saranoita sekä päällekkäisten pieli- tai pilarielementtien välistä vaakaliitosta. Päällekkäisten pieli- tai pilarielementtien vaakaliitoksen mallintaminen on helppoa ja tehokasta perinteisellä menetelmällä, joten tässä diplomityössä ei tutkita niiden algoritmiavusteista mallinnusta.

Parvekeputket mallinnetaan Sweco Rakennetekniikka Oy:n kehittämällä Tekla pluginillä ja parvekesaranat mallinnetaan Peikko Finland Oy:n tai Anstar Oy:n kehittämällä Teklan custom componenteilla. Tässä tapauksessa custom componentilla tarkoitetaan Tekla ympäristöön tehtyjä valmiita osia, esimerkiksi elementtien valutarvikkeita. Kuvassa 29 on Peikon PS 200 parvekesaranan custom component Tekla ympäristössä.



Kuva 29. Peikon PS 200 parvekesaranan custom component Tekla ympäristössä.

Parvekeputkien ja -saranoiden mallinnusta varten parvekelaatat eroteltiin ryhmiin sen mukaan, tuleeko niihin putkea vai ei. Tämän jälkeen tehdään algoritmi, joka tunnistaa laatoista rakennuksen sisäpuoleisen reunan, eli sen reunan johon parvekeputket ja -saranoit tulee. Parvekeputkien osalta seuraavaksi tehdään algoritmi, joka hakee laatan sisäpuoliselta reunalta nurkkapisteen, joka on lähimpänä pilaria tai pilarimaista pieltä. Lisäksi määritetään kiinteä parametri, joka siirtää pisteen sopivalle etäisyydelle laatan nurkasta. Parvekeputken Tekla plugin tarvitsee toimiakseen pääosan sekä kaksi pistettä, joilla määritetään putken sijainti ja suunta. Edellä kuvatun algoritmin lopputuloksena on ensimmäinen piste, joten tarvitaan vielä toinen piste. Toinen piste saadaan, kun muodostetaan algoritmi, joka hakee ensimmäistä pistettä vastaavan pisteen kohtisuoraan laatan toiselta puolelta. Kun pisteet on eroteltu, muodostetaan algoritmi, joka monistaa

pisteet oikeisiin korkoihin. Lopuksi muodostetaan Component Input -komponentilla Component -komponentille oikea tietojen syöttämisen järjestys ja mallinnetaan parvekeputket.

Parvekesaranoiden osalta muodostetaan algoritmi, joka lisää kaikkiin parvekelaattoihin kaksi kappaletta parvekesaranoita neljännespisteisiin. Jos parvekelaatassa on putki, poistetaan putkea lähinnä oleva neljännespiste. Custom componentin mallintamiseen Component -komponentilla tarvitaan kaksi pistettä. Ensimmäinen piste on neljännespiste ja toinen piste saadaan vastaavalla tavalla kuin parvekeputkella.

Kun parvekeputket ja -saranat on saatu mallinnettua, on vuorossa muu detaljointi ja raudoitus. Muulla detaljoinnilla tarkoitetaan valutarvikkeiden mallinnusta. Parvekelaatoille tyypillisiä valutarvikkeita ovat nostoankkurit, parvekekaivot, sisäkierreosa laatan pohjassa turvalajaita varten sekä sisäkierreosat laatan yläpinnassa pielien ja pilarien tönäreitä varten. Parvekepielille ja pilareille tyypillisiä valutarvikkeita ovat nostolenkit tai -ankkurit, elementtien välisten liitosten vaatimat osat sekä sisäkierreosat tönäreille. Teklassa näiden kaikkien edellä lueteltujen valutarvikkeiden mallinnus perinteisellä menetelmällä on melko yksinkertaista ja tehokasta, joten tässä diplomityössä ei tutkita niiden algoritmiavusteista mallinnusta.

Parvekelaattojen tyypillisiä raudoitteita ovat ympäri kiertävät pieliteräkset, verkot ylä- ja alapinnassa sekä reunahaat. Parvekepielissä tyypillisesti on vain ympäri kiertävät pieliteräkset sekä aukkopalkkien haat. Verkkoraudoitteita ei tyypillisesti pielissä tarvita paitsi silloin kuin pieli on altis törmäyskuormille. Pilarien raudoitus lasketaan aina tapauskohtaisesti. Laattojen, pielien ja pilarien raudoitteiden mallintaminen on tehokasta valmiilla Tekla plugineilla jo perinteisellä menetelmällä, joten tässä diplomityössä ei tutkita niiden algoritmiavusteista mallinnusta.

Detaljoinnin jälkeen suunnitelmat tarkastetaan ja mahdolliset virheet korjataan. Kun tarkastus on hyväksytty, tehdään elementtien valmistuspiirustukset. Kuvatuotannon nykyiset prosessit ja kuvapohjat ovat kehittyneet todella pitkälle, joten niiden uudelleen kehittäminen algoritmiavusteisesti ei ole järkevää. Tästä syystä algoritmiavusteista kuvatuotantoa ei tässä diplomityössä tutkita.

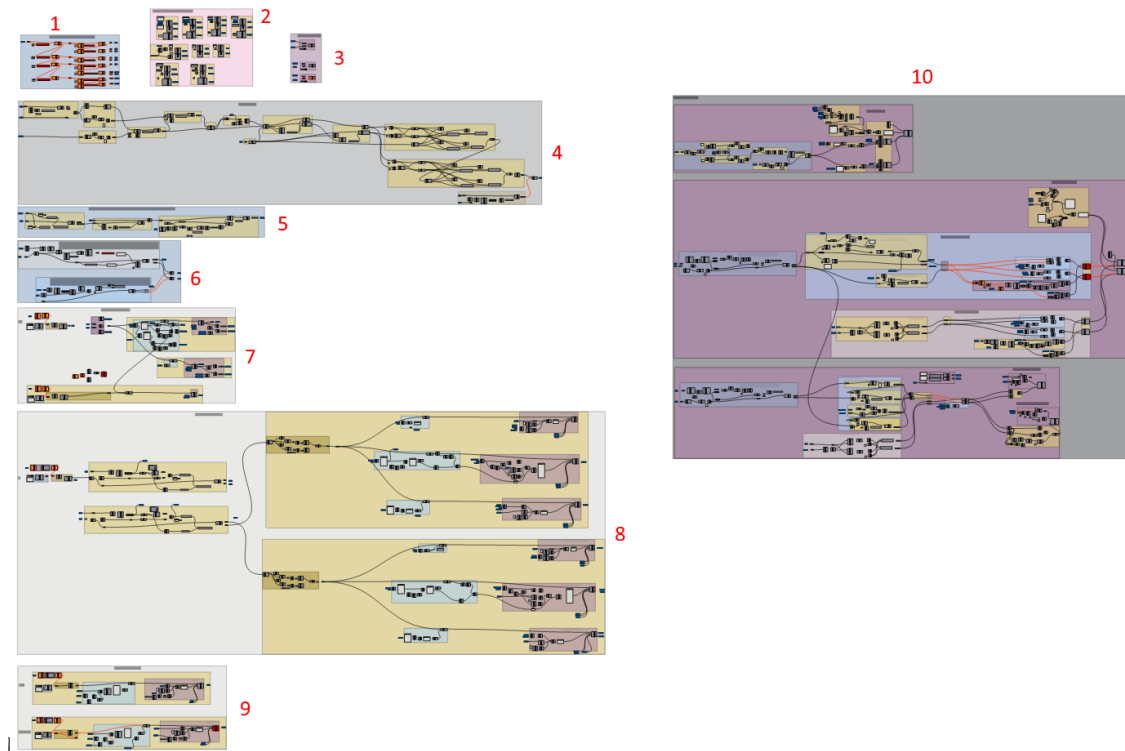
5.3 Case-tutkimuksessa luotu algoritmi

Tässä kappaleessa käsitellään case-tutkimukseen luotua algoritmia. Algoritmi on laaja kokonaisuus ja se muodostuu useista osioista. Pilkotaan algoritmi eri osiin ja käsitellään niiden toimintaa. Tarkoitus ei ole kuvata algoritmia yksityiskohtaisesti, vaan esitellä algoritmin toimintaperiaate yleisellä tasolla.

5.3.1 Algoritmin ryhmittely, jäsentely ja käyttöliittymä

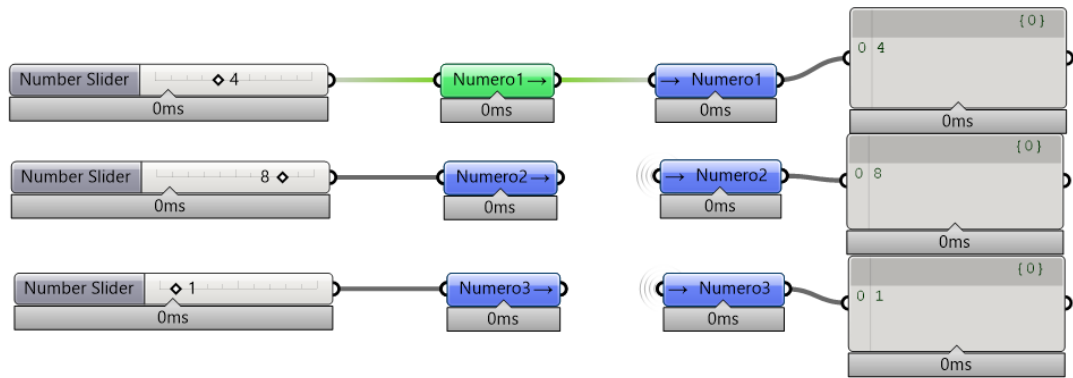
Aiemmin teoria osuudessa, kappaleessa kaksi, perehdyttiin algoritmin muodostamiseen, jäsentelyyn, ryhmittelyyn ja uudelleen hyödynnettävyyteen. Näitä oppeja pyrittiin hyödyntämään algoritmia luotaessa. Algoritmin luomisessa hyödynnettiin hajautta ja hallitse -tekniikkaa. Kuten aiemmin kappaleessa 5.2.4 mainittiin, algoritmin toiminta kirjoitettiin sanalliseen muotoon, jaettiin osakokonaisuuksiin ja ratkaistiin ne. Kuvassa 30 on esitetty osa case-tutkimukseen luodusta algoritmista. Kuvasta huomataan, algoritmi on hyvin laaja kokonaisuus. Algoritmin numeroidut osat ovat:

1. Teklaan mallinnetun geometrian lähtötiedon erottelu
2. Algoritmin lopputuloksena mallinnettujen parveke-elementtien tiedot
3. Elementtien aukotukset
4. Laattojen tukipinnan korjaus
5. Laattojen erottelu parvekeputkien ja saranoiden mukaan
6. Laattojen erottelu parvekeputkien ja saranoiden mukaan
7. Parvekelaatan, laatan aukotusten sekä kattolaatan mallinnus
8. Parvekepielien mallinnus
9. Ikkuna- ja oviaukkojen mallinnus
10. Valulippojen leikkaus sekä parvekeputkien ja -saranoiden mallinnus.



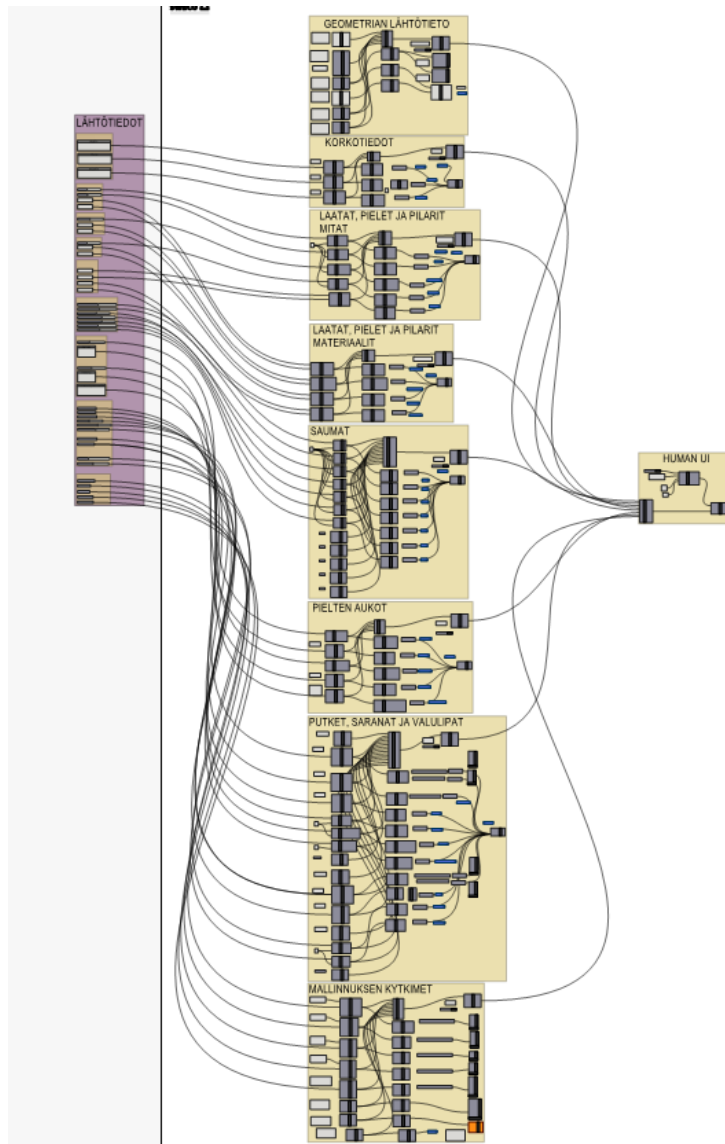
Kuva 30. Osa case-tutkimukseen luodusta algoritmista.

Kuvasta 30 huomataan myös käytetty hajauta ja hallitset -tekniikka. Algoritmin osakoko-
naisuudet ovat ryhmitelty Grasshopperin Group -toiminnolla. Ryhmien värit ovat jaoteltu
kyseisen algoritmin toiminnallisuuden mukaan. Värien luomisessa on hyödynnetty
Grasshopperin lisäosia Palette ja MetaHopper. Paletella voidaan muokata Grasshoppe-
rin värimaailmaa hyvin monipuolisesti. MetaHopperilla voidaan ohjata Grasshopperin
komponentteja dynaamisesti. Algoritmeissa on hyödynnetty myös hyvin paljon
Grasshopperin lisäosaa nimeltään Telepathy. Telepathy yksinkertaisuudessaan sisältää
vain kaksi komponenttia: Remote Sender ja Remote Receiver. Telepathyllä voidaan kor-
vata eri komponenttien välisiä lankoja. Sen toiminta perustuu komponentille asetetta-
vaan nimeen. Esimerkki Telepathyn toiminnasta on esitetty kuvassa 31. Kuvassa on
kolme Number Slider -komponenttia, joissa jokaisessa on eri numero. Number Sliderit
on yhdistetty Remote Sender -komponentteihin, joille jokaiselle on annettu oma nimi.
Remote Sender lähettää siihen liitetyn tiedon, eli tässä tapauksessa numeron, saman
nimiseen Remote Receiveriin ilman lankaa. Kuvassa ylin Remote Sender on aktivoitu,
jolloin se näyttää langalla sitä vastaavan Remote Receiverin. Telepathyn avulla on hyvin
helppoa ja vaivatonta vähentää lankojen määrää ja tällä tavalla selkeyttää algoritmia
merkittävästi.



Kuva 31. Esimerkki Telepathy lisäosan toiminnasta.

Algoritmin käyttöä varten luotiin käyttöliittymä Grasshopperin Human UI lisäosalla. Human UI:llä voidaan luoda erilliseen ikkunaan käyttöliittymä, jolla voidaan käyttää koko algoritmia yhtä poikkeusta lukuun ottamatta. Kuvissa 32 ja 33 on esitetty käyttöliittymän algoritmi Grasshopperissa sekä itse käyttöliittymän ikkuna. Kuvassa 32 algoritmin vaatimat lähtötiedot ovat vasemmalla, käyttöliittymän osiot keskellä ja käyttöliittymän ikkunan muodostava komponentti oikealla. Käyttöliittymän ikkuna on esitetty seuraavassa kuvassa. Käyttöliittymässä eri osiot ovat jaoteltu omiin ryhmiin Group -toiminnolla.



Kuva 32. Lähtötiedot sekä Human UI käyttöliittymän algoritmi.

Kuvassa 33 on esitetty käyttöliittymän ikkuna. Käyttöliittymän ensimmäisessä osiossa tuodaan geometrian DWG-tiedosto Rhinoon, seuraavassa määritellään anturan, lattian sekä kattolaatan korot, seuraavassa rakenneosien paksuudet sekä profiilit ja viimeisessä avoimessa osiossa rakenneosien materiaalit. Teklassa mallintamalla luotavaa geometrian lähtötietoa ei saatu sisällytettyä käyttöliittymään johtuen Human UI:n sekä Grasshopper-Tekla Live Linkin rajoitteista.

The screenshot shows the 'PARVEKETORNI' application window. It has a blue header bar with the title 'PARVEKETORNI'. Below the header, there are several sections with expandable/collapsible headers:

- GEOMETRIAN LÄHTÖTIEDOT**: Contains two buttons: 'Tuo DWG tiedosto' and 'Poista DWG tiedosto'.
- KORKOTIEDOT**: Contains three input fields:
 - 'Anturan YP+ korko' with value '100730'
 - 'ARK+ korot' with value '101840,104840,107840,110840,113840'
 - 'Kattolaatan YP+ korko' with value '116800'
- LAATAT, PIELET JA PILARIT MITAT**: Contains four sliders and one input field:
 - 'Laatan paksuus' slider set to 270
 - 'Laatan tukipinta' slider set to 60
 - 'Kattolaatan paksuus' slider set to 270
 - 'Pielen paksuus' slider set to 200
 - 'Pilarin profiili' input field with value 'D150'
- LAATAT, PIELET JA PILARIT MATERIAALIT**: Contains four input fields:
 - 'Laatan materiaali' with value 'C30/37 SÄÄNKESTÄVÄ'
 - 'Kattolaatan materiaali' with value 'C30/37 SÄÄNKESTÄVÄ'
 - 'Pielen materiaali' with value 'C25/30'
 - 'Pilarin materiaali' with value 'S235JR'

At the bottom, there is a list of expandable sections:

- SAUMAT
- PIELTEN AUKOT
- PARVEKEPUTKET, -SARANAT JA VALULIPAT
- MALLINNUKSEN KYTKIMET

Kuva 33. Algoritmin käyttöliittymän ikkuna.

5.3.2 Algoritmin lähtöparametrit ja yksinkertaistukset

Algoritmile syötetään seuraavat lähtöparametrit:

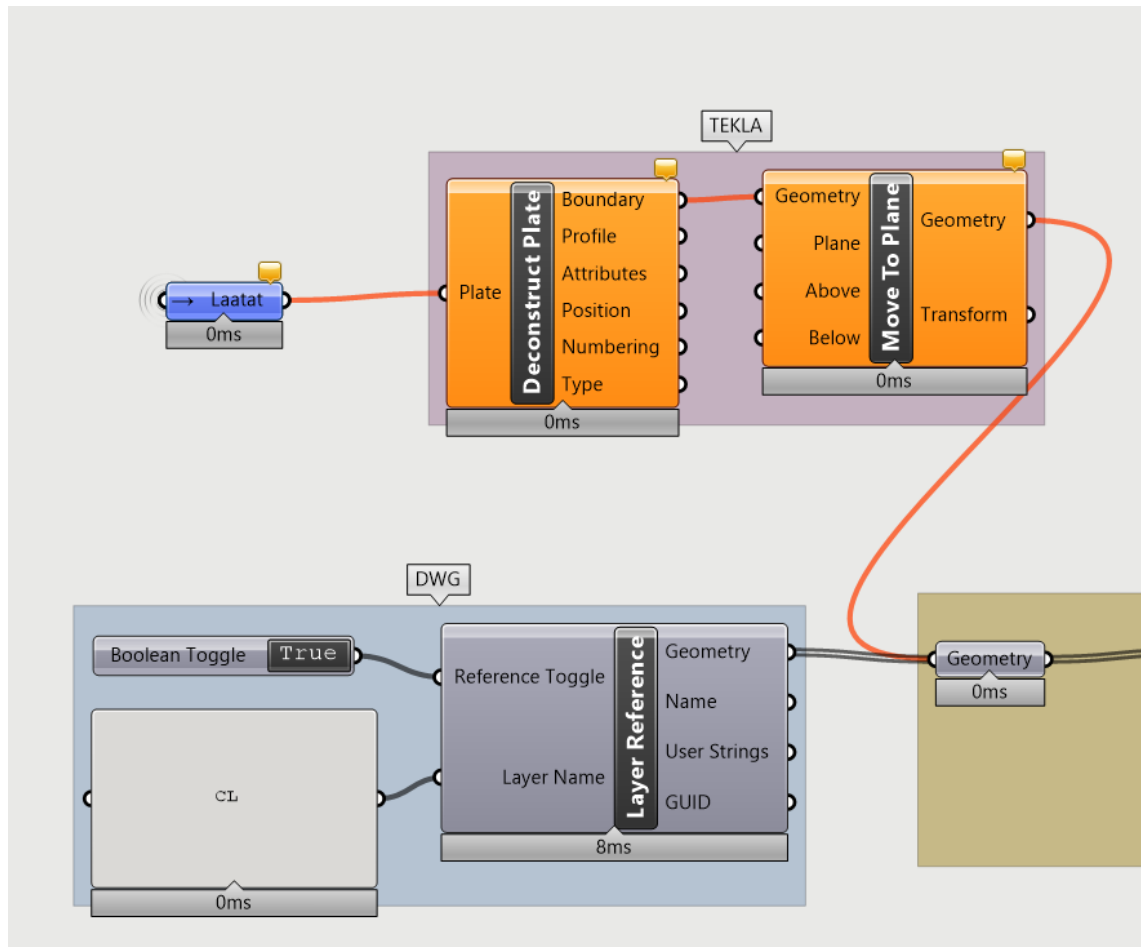
- Korkotiedot
 - Anturan YP+
 - Lattian YP+
 - Kattolaatan YP+
- Parvekelaatta
 - Paksuus
 - Tukipinta (myös katto-laatta)
 - Materiaali
- Parvekekattolaatta
 - Paksuus
 - Materiaali
- Parvekepieli
 - Paksuus
 - Materiaali
- Parvekepilari
 - Profiili
 - Materiaali
- Elementtien välisten saumojen paksuudet
 - Laatan ja pielen saumat
 - Laatan ja pilarin saumat
 - Laatan etäisyys lattian YP+
 - Anturan ja pielen sauma
 - Anturan ja pilarin sauma

- Aukotukset
 - Kerrosten määrä, joissa oviaukko
 - Oviaukon korkeus
 - Kerrosten määrä, joissa ikkuna-aukko
 - Ikkuna-aukon korkeus
 - Ikkuna-aukon korkeus pielen alareunasta
- Parvekeputket
 - Mallinnus, automaattinen tai manuaalinen
 - Pituus
 - Profiili
 - Ulostulo
 - Korkeus laatan pohjasta
 - Etäisyys laatan nurkasta
- Parvekesarana
 - Mallinnus, automaattinen tai manuaalinen
 - Saranan tyyppi, Peikko tai Anstar
- Valulipat
 - Lipan paksuus

Käytettävistä parametreista johtuen algoritmiin on pitänyt tehdä tiettyjä yksinkertaistuksia. Rakenteiden paksuudet ja profiilit ovat vakioita. Algoritmilli ei voida mallintaa eri paksuisia rakenteita, vaan paksuuksien vaihtelut pitää muuttaa manuaalisesti algoritmin mallinnuksen jälkeen. Toinen merkittävä yksinkertaistus on, että algoritmi mallintaa jokaisen kerroksen samalla tavalla. Kerros kohtaista parvekkeiden vaihtelua algoritmilli ei voida tehdä.

5.3.3 Geometrian kokoaminen

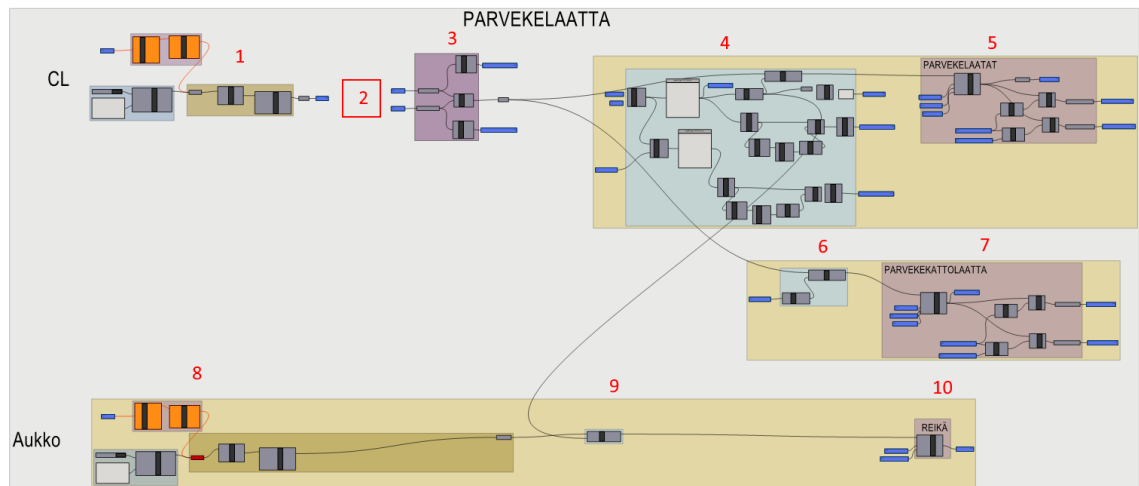
DWG-tiedostoon piirretty tai Teklaan mallinnettu geometrian lähtötieto pitää muuttaa Rhinon ja Grasshopperin ymmärtämään muotoon. Muutos tapahtuu kuvan 34 mukaisella algoritmilli. Kuvassa ylempänä on Teklaan mallinnetun parvekelaatan geometrian muuttaminen. Grasshopper-Tekla Live Linkin Deconstruct Plate -komponentilla saadaan Teklan Slab -komponentilla mallinnetun laatan geometria, joka vielä siirretään XY-tasoon. DWG-tiedostoon määritelty layer luetaan Rhinosta Layer Reference -komponentilla. Komponentille määritetään käytettävän layerin nimi.



Kuva 34. Geometrian muuttaminen Rhinon ja Grasshopperin ymmärtämään muotoon.

5.3.4 Parvekelaatta

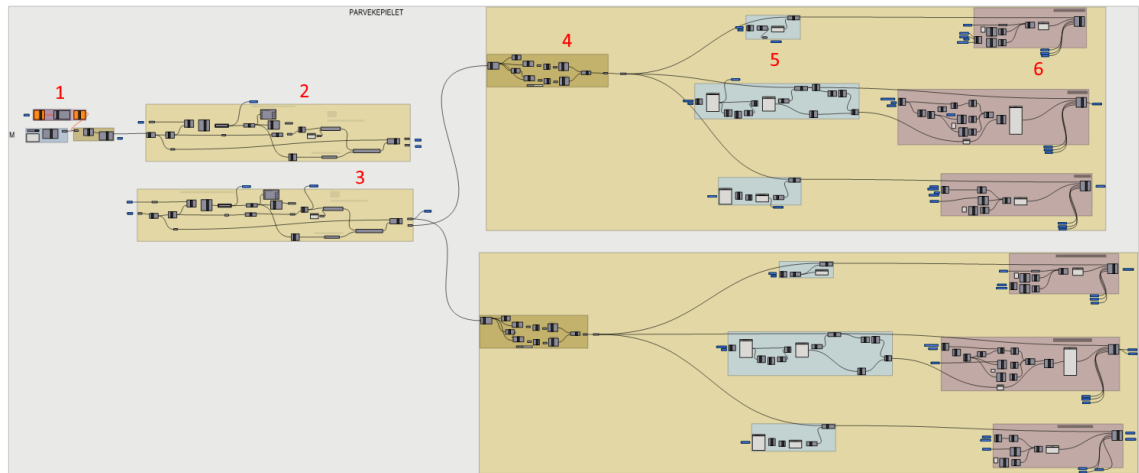
Kuvassa 35 on esitetty parvekelaatan, laatan aukkojen sekä kattolaatan mallinnuksen algoritmit. Kuvaan on merkattu numeroilla eri osa-alueet. Ensimmäisessä osassa on kuvassa 34 esitetty geometrian muutos, toisessa osassa on laattojen tukipinnan korjaus, jonka algoritmi ei kuvassa näy. Kolmannessa osassa yhdistetään laatat, joissa on parvekeputki ja laatat, joissa putkea ei ole. Neljännessä monistetaan geometria oikeisiin korkoihin ja viidennessä mallinnetaan parvekelaatat. Kuudennessa siirretään kattolaatan geometria oikean korkoon ja seitsemännessä mallinnetaan se. Kahdeksas, yhdeksäs ja kymmenes on laatan aukon geometrian muutos, monistus ja mallinnus.



Kuva 35. Parvekelaatan, laatan aukkojen sekä kattolaatan mallinnuksen algoritmit.

5.3.5 Parvekepieli

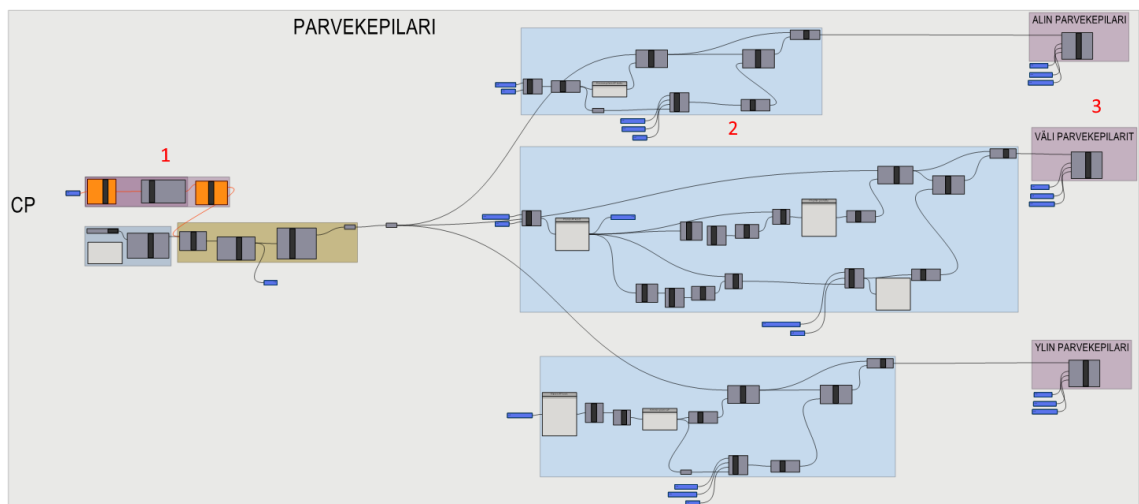
Kuvassa 36 on esitetty parvekepielten mallinnuksen algoritmit. Ensimmäisessä osassa on pielen geometrian muuttaminen Rhinon ja Grasshopperin muotoon. Toisessa osassa on reuna- ja keskapielten erottelun algoritmi. Erottelu tapahtuu törmäyttämällä laatat ja pielet keskenään. Jos yhteen pieleen törmää kaksi laattaa, pieli on keskapieli. Jos pieleen törmää vain yksi laatta, verrataan pielen poikkileikkauksen pinta-alaa sekä laatan ja pielen törmäyspinta-alaa keskenään. Jos pinta-alojen suhde on alle 0.8, pieli on reunapieli. Toisen osion jälkeen tehdään laattojen tukipintojen korjaus, jonka jälkeen kolmannessa osassa tarkastetaan vielä pielten erottelu. Neljännessä osassa haetaan pielen poikkileikkauksen keskilinja. Viidennessä siirretään ja monistetaan geometria oikeisiin korkoihin ja kuudennessa mallinnetaan pielet. Mallinnus on jaoteltu kolmeen osaan: ensin alin pieli, lopuksi ylin pieli ja näiden välissä muut pielet. Reuna- ja keskapiellet on vielä erotettu toisistaan keskapielten ollessa oikealla ylempänä ja reunapiellet alempana. Ovi- ja ikkuna-aukkojen mallinnus noudattaa samaa logiikkaa ilman osia kaksi ja kolme.



Kuva 36. Parvekepielten mallinnuksen algoritmit.

5.3.6 Parvekepilari

Kuvassa 37 on esitetty parvekepilarien mallinnuksen algoritmit. Ensimmäisessä osassa on pilarin geometrian muuttaminen Rhinon ja Grasshopperin muotoon sekä pilarin poikkileikkauksen keskipisteen muodostavat algoritmit. Toisessa osassa ensin siirretään ja monistetaan pilarin alapinta oikealle korolle. Tämän jälkeen lasketaan pilarin korkeus ja monistetaan sille toinen piste. Viimeisessä osassa mallinnetaan pilari näiden pisteiden mukaan. Osat kaksi ja kolme ovat jaettu kolmeen eri osaan alimmalle pilarille, välipilarille sekä ylimmälle pilarille.

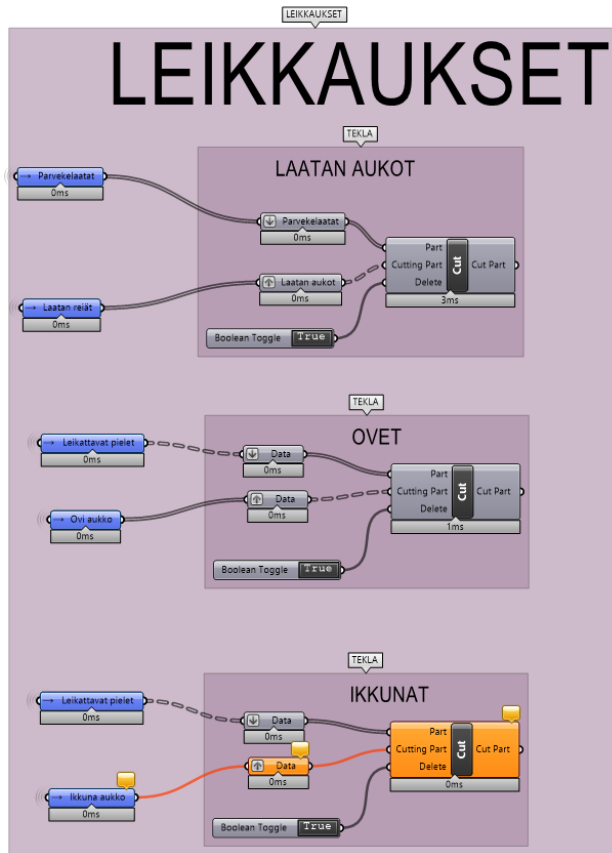


Kuva 37. Parvekepilarin mallinnuksen algoritmit.

5.3.7 Detaljointi

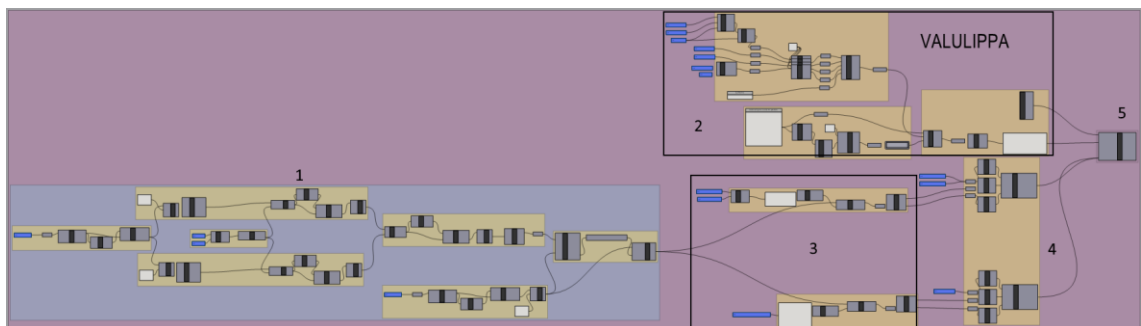
Detaljoinnin ensimmäinen vaihe oli aukotusten tekeminen. Edellisessä kappaleessa esiteltiin eri elementtien ja aukotusten mallintamisen algoritmit. Kuvassa 38 on esitetty aukkojen leikkausten algoritmit. Ylimpänä on laatan aukkojen leikkaus. Cut -komponenttiin

yhdistetään ensin mallinnetut laatat, seuraavaksi mallinnetut aukot ja lopuksi poistetaan leikkauskappale leikkauksen jälkeen. Ovi- ja ikkuna-aukkojen leikkaukset tehdään samalla tavalla.



Kuva 38. Aukkojen leikkausten algoritmit.

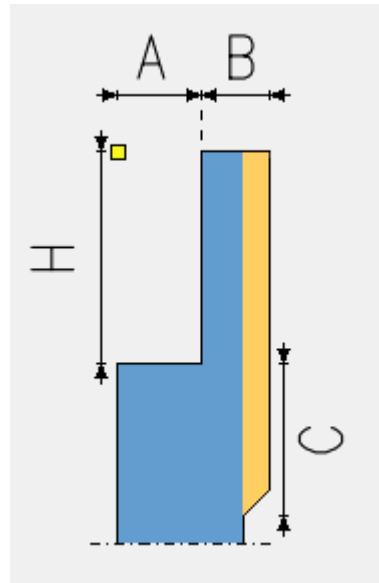
Seuraavaksi leikattiin reunapieliin valulipat. Kuvassa 39 on esitetty valulippojen leikkauksen algoritmit. Kuvassa ensimmäisenä on algoritmi, joka etsii pienen poikkileikkauksesta parvekelaatan puoleisen reunan. Toisessa osassa määritetään valulipan leikkauksen tekävän Tekla pluginin asetukset.



Kuva 39. Valulippojen leikkauksen algoritmit.

Valulipalle määritetään kuvan 40 mukaiset mitat sekä leikkauskappaleen nimi. Algoritmi määritetään lähtötietona lipan paksuus B sekä ylimääräisen vahvikkeen korkeus

C. Algoritmi laskee leikkauksen leveyden A ja korkeuden H laatan paksuuden, laatan tukipinnan, lipan paksuuden sekä elementtien välisten sauma-arvojen mukaan.

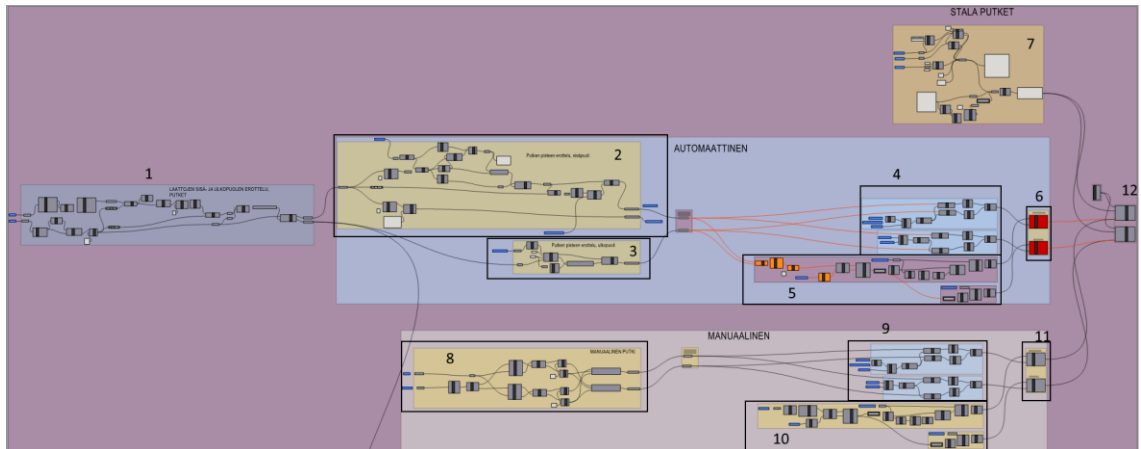


Kuva 40. Valulipan leikkaavan Tekla pluginiin määritettävät parametrit.

Kuvan 39 kolmannessa osassa monistetaan ensimmäisen osan pielen reuna oikeisiin korkoihin. Neljännessä osassa muodostetaan pääosan, alkupisteen ja loppupisteen syöttäminen oikealla tavalla Component Input -komponentilla. Viidennessä osassa on Component -komponentti, johon on yhdistetty tarvittavat tiedot ja joka suorittaa leikkaavan pluginin.

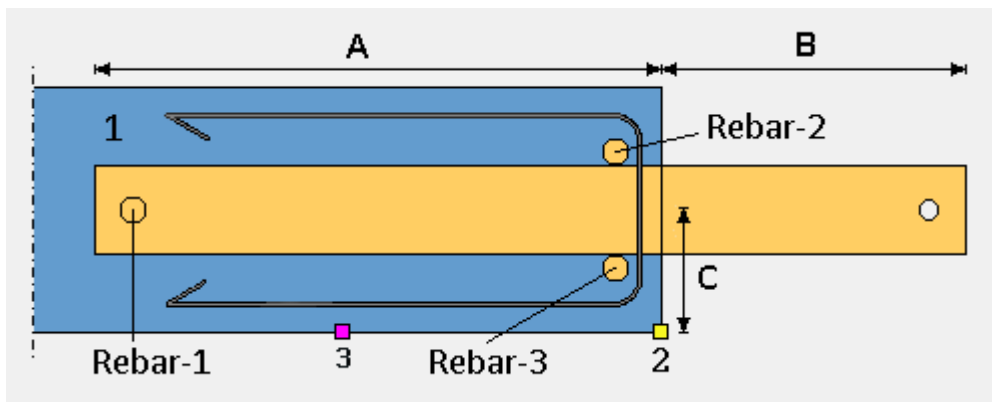
Valulippojen leikkausten jälkeen vuorossa oli parvekeputkien mallinnus. Kuvassa 41 on esitetty parvekeputkien mallinnuksen algoritmit. Algoritmi on jaettu 12 osaan ja ensimmäisenä on algoritmi, joka erottelee parvekelaatoista rakennuksen sisä- ja ulkopuoliset reunat. Putkien mallinnukseen on kaksi tapa: automaattinen tai manuaalinen. Automaattisella tavalla algoritmi päättää itsenäisesti ne laatat, joihin parvekeputket tarvitaan. Manuaalisella tavalla käyttäjä on itse määrittänyt parvekeputkien paikat joko DWG-tiedostoon tai Teklaan. Toisessa osassa algoritmi etsii laatan sisäpuoleisesta reunasta sen päädyn, joka on lähimpänä parvekepilari tai pilarimaista pieltä. Algoritmilta syötettävällä parametrilla voidaan säätää putken pisteen etäisyyttä parvekelaatan nurkasta. Kun oikea piste on löydetty, kolmannessa osiossa algoritmi etsii pistettä kohtisuoraa vastaavan pisteen laatan toiselta puolelta. Kahdeksannessa osassa algoritmi etsii vastaavat pisteet kuin toisessa ja kolmannessa osassa käyttäjän määäämistä sijainnista. Neljäs ja yhdeksäs osa monistavat pisteet oikeisiin korkoihin. Viides ja kymmenes osa korjaa parvekelaattojen lukumäärän vastaamaan putkien lukumäärää. Kuudes ja yhdestoista osa

muodostaa pääosan ja pisteiden oikean syöttämisen Component -komponentille. Seitsemäs osa määrittelee parvekeputken Tekla pluginille syötettävät parametrit. Kahdestoistaosa mallintaa parvekeputket.



Kuva 41. Parvekeputkien mallinnuksen algoritmit.

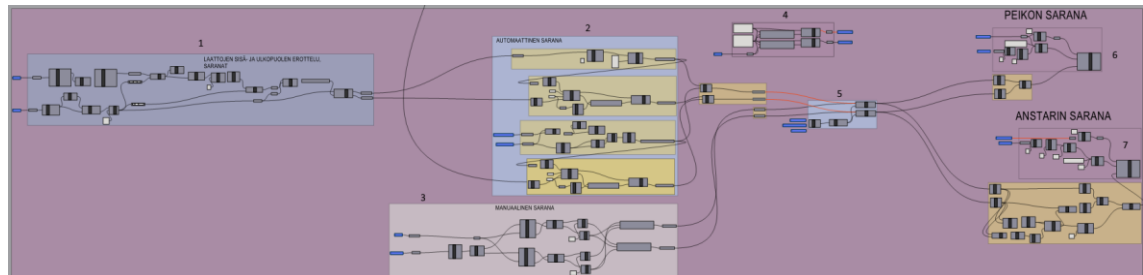
Algoritmile määritetään kuvan 42 mukaiset kiinteät parametrit putken ulostulo B ja putken korkeus laatan pohjasta C. Lisäksi määritetään putken materiaali ja profiili. Kuvassa näkyviä raudoituksia Rebar-1, -2 ja -3 ei algoritmiin määritellä, koska algoritmi ei suorita parvekeputken kapasiteetin laskentaa.



Kuva 42. Parvekeputket mallintava Teklan plugin.

Parvekesaranoiden mallinnuksen algoritmit ovat esitetty kuvassa 43. Parvekesaranoiden mallinnuksen logiikka on samankaltainen kuin parvekeputkien. Ensimmäisessä osassa erotellaan laatan sisä- ja ulkopuoliset reunat. Saranoiden mallinnukseen on myös kaksi vaihtoehtoa: automaattinen ja manuaalinen. Automaattisella tavalla toisessa osassa algoritmi etsii laatoista neljännespisteet. Jos laatasta on putki, putkea lähin neljännespiste poistetaan. Manuaalisella tavalla kolmannessa osassa algoritmi etsii mallinnukseen tarvittavat pisteet käyttäjän määrittämästä kohdasta. Neljännessä osassa erotellaan, onko kyseessä Peikon vai Anstarin sarana. Eri valmistajien saranat pitää erottaa toisistaan, koska ne mallinnetaan omilla custom komponenteilla. Viidennessä osassa

monistetaan pisteet oikeille koroille. Kuudes osa mallintaa Peikon saranat ja seitsemäs osa Anstarin saranat.



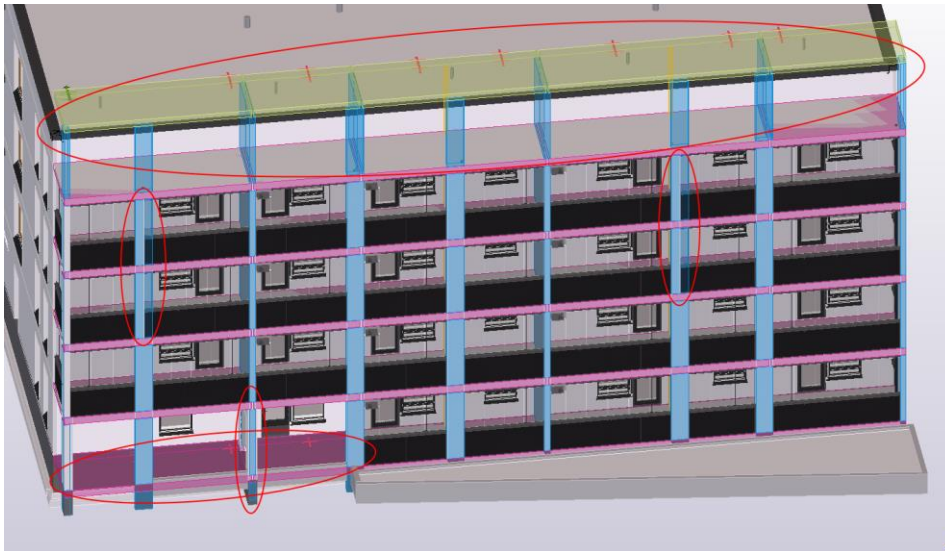
Kuva 43. Parvekesaranoiden mallinnuksen algoritmit.

6. CASE-TUTKIMUKSEN TULOSTEN ANALYSOINTI

Kuten aiemmin mainittua, tämän diplomityön voidaan ajatella olevan jatkoa Erkkilän (2017) diplomityölle. Vertaillaan tässä kappaleessa algoritmin suorittamaa mallinnusta case-kohteen arkkitehdin IFC-malliin. Käsitellään myös case-tutkimuksessa havaittuja ongelmia sekä pohditaan niiden syitä ja mahdollisia ratkaisuja. Suoritetaan myös vertailua Erkkilän diplomityössä havaittuihin ongelmiin. Vertailulla havaitaan kuinka käytettävät ohjelmistot ovat kehittyneet sekä kuinka geometrian lähtötiedon eroavaisuus vaikuttaa suunnitteluprosessiin.

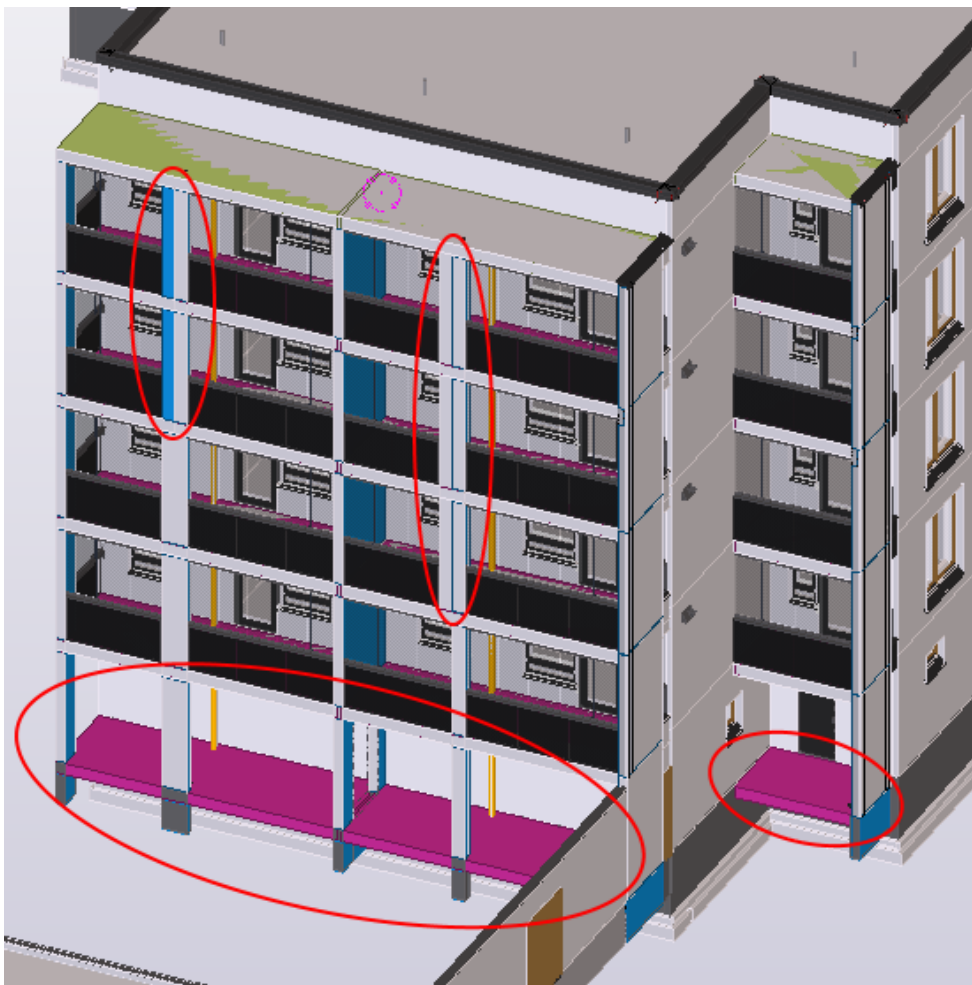
6.1 Case-tutkimuksen tulosten vertailu toteutuneeseen kohteeseen

Tässä kappaleessa vertaillaan algoritmilla mallinnettuja parveke-elementtejä kohteen arkkitehdin IFC-malliin. Aiemmin case-tutkimuksen alussa kappaleessa 5.2.1 esiteltiin case-tutkimuksen kohde. Kohteesta osa on neljän ja osa viiden kerroksen korkuinen. Neljä kerrosta korkeassa osassa parvekkeet alkavat ensimmäisestä kerroksesta kahta parvekettä lukuun ottamatta. Viisi kerrosta korkeassa osassa kaikki parvekkeet alkavat toisesta kerroksesta. Tästä parvekkeiden vaihtelusta ja aiemmin kappaleessa 5.3.2 esitetyistä algoritmin yksinkertaistuksista johtuen kohteen kaikkia parvekkeita ei saada mallinnettua algoritmilla kerralla. Kuvissa 44 ja 45 on esitetty kaikki kerralla mallinnetut parveke-elementit sekä arkkitehdin IFC-malli. Kuvasta 44 huomataan punaisella merkatuilta alueilta, että ensimmäiseen kerrokseen tulee kaksi ylimääräistä parvekelaattaa, katto-laatat ovat yhden kerroksen liian korkealla ja algoritmi ei huomioi pilarimaisen pielen leveyden vaihtelua eikä pielen paksuuden vaihtelua.



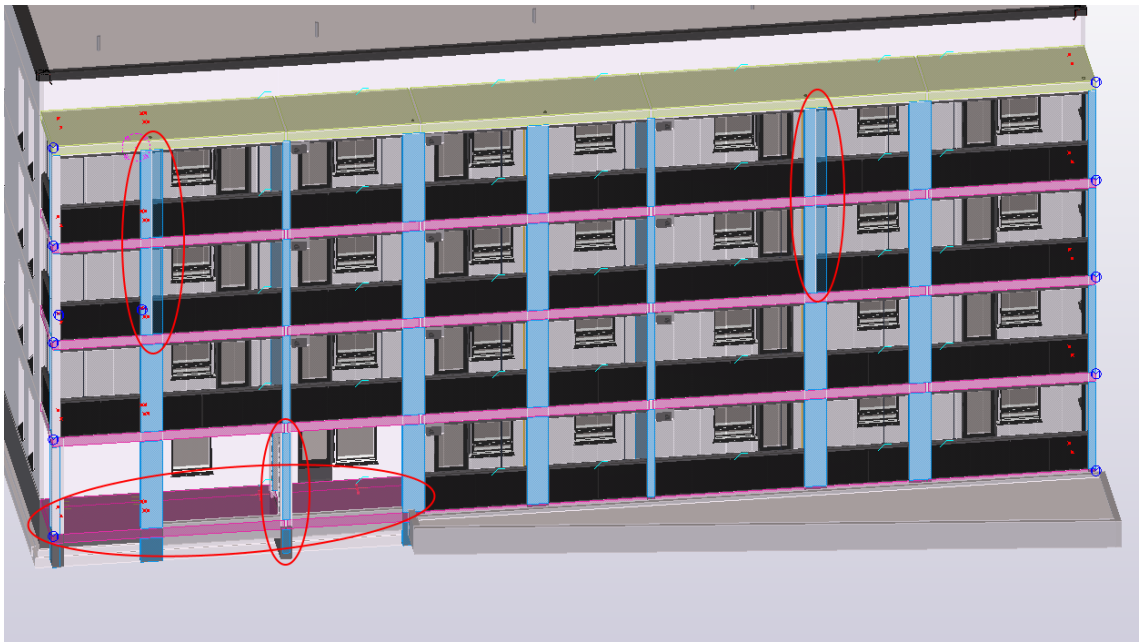
Kuva 44. Kaikki kerralla algoritmilla mallinnetut parvekkeet ja arkkitehdin IFC-malli.

Kuvasta 45 huomataan punasilla merkityiltä alueilta, että ensimmäiseen kerrokseen tulee ylimääräiset parvekelaatat sekä algoritmi ei huomioi pilarimaisen parvekepielen leveyden vaihtumista.



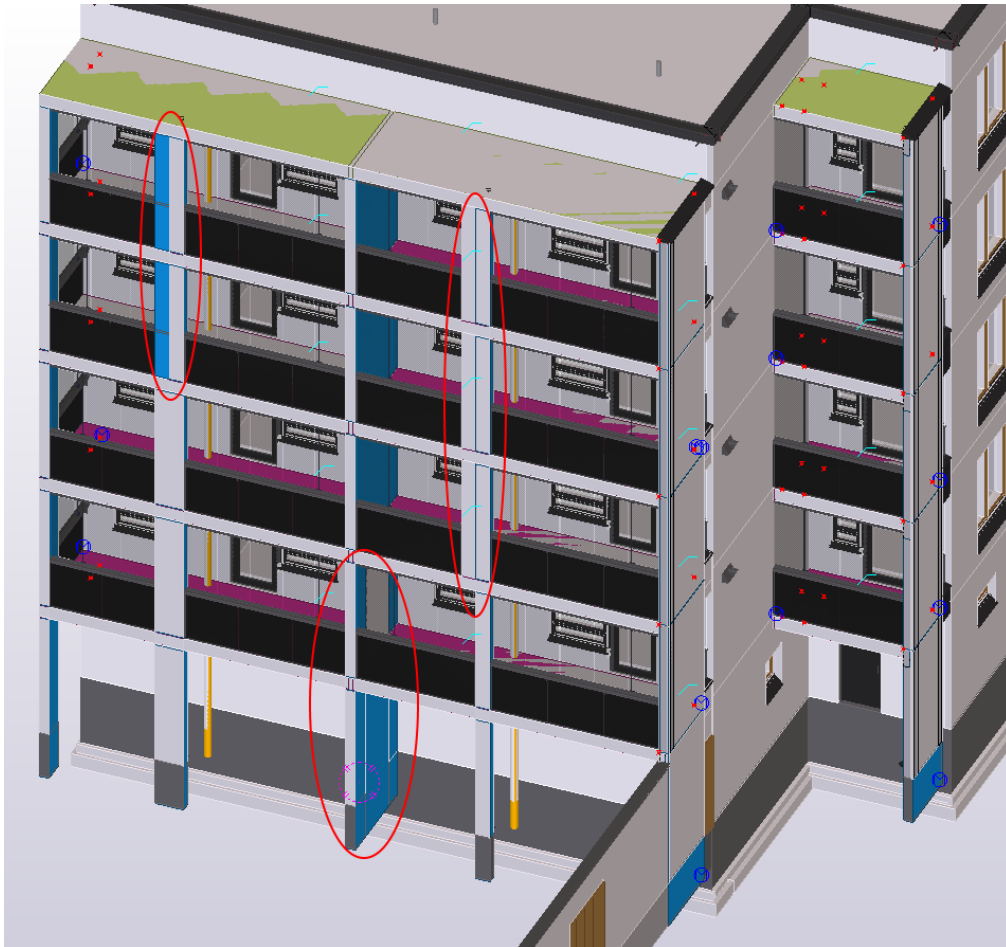
Kuva 45. Kaikki kerralla algoritmilla mallinnetut parvekkeet ja arkkitehdin IFC-malli.

Kuvien 44 ja 45 ylimääräisten parvekelaattojen ongelmat saadaan ratkaistua mallintamalla algoritmilla eri korkuiset parvekelinjat erikseen. Kuvissa 46 ja 47 on esitetty erikseen algoritmilla mallinnetut neljän ja viiden kerroksen korkuiset parvekelinjat. Kuvasta 47 huomataan, että mallintamalla algoritmilla vain neljä kerrosta korkeat parvekkeet kerralla saadaan ratkaistua kattolaattojen ongelma.



Kuva 46. Neljän kerroksen korkuiset parvekkeet ja arkkitehdin IFC-malli.

Kuvasta 47 huomataan, että mallintamalla algoritmilla vain viisi kerrosta korkeat parvekkeet kerralla saadaan ratkaistua ensimmäisen kerroksen parvekkeiden ongelma. Kuvasta huomataan myös, että tällä tavalla mallintaen ilmenee uusi ongelma. Alimman parvekepielen oviaukko mallintuu yhden kerroksen liian korkealle. Ongelman saisi ratkaistua muuttamalla algoritmia.



Kuva 47. Viiden kerroksen korkuiset parvekkeet ja arkkitehdin IFC-malli

Todettakoon, että yllä olevissa kuvissa merkatut parvekepielten leveyden tai paksuuden muutokset ovat helpot toteuttaa mallissa jo nykyisellä menetelmällä. Tämän takia algoritmiin tehtiin yksinkertaistus, joka ei salli paksuuden tai leveyden vaihtelua.

6.2 Algoritmiavusteisen suunnittelun ongelmat

Perehdytään algoritmiavusteisen suunnittelun ongelmiin, joita ilmeni case-tutkimuksen yhteydessä. Vertaillaan myös havaittuja ongelmia Erkkilän (2017) havaintoihin. Käytetään samoja otsikoita kuin Erkkilä käytti omassa diplomityössään.

6.2.1 Algoritmia ja mallintamista edeltävät ongelmat

Erkkilä käytti tutkimuksessaan väliseinäelementtien geometrian lähtötietona arkkitehdin IFC-mallia. Hän havaitsi tutkimuksessaan kolme keskeistä ongelmaa: arkkitehdin mallinnusvirheet, arkkitehdin IFC-mallin tietovirheet ja tarkastuksen haastavuus. Arkkitehdin mallinnusvirhettä voidaan pitää vakavana ongelmana, sillä sen havaitseminen liian myöhään voi aiheuttaa suuria ongelmia elementtisuunnittelussa. IFC-mallin tietovirheet ovat

myös vakava ongelma, koska ne hankaloittavat väliseinien erottelua IFC-mallista. Pahimmassa tapauksessa jotkin seinistä saattavat jäädä kokonaan pois erottelusta. Tarkastuksen haastavuus liittyy käytettyyn SimpleBim-ohjelmistoon. SimpleBim ei sovellu visuaaliseen tarkastukseen kovinkaan hyvin ja käytännön tarkastus pitää tehdä Teklasta, jonka visuaalisuus on parempi. (Erkkilä 2017, s. 80-81)

Tässä tutkimuksessa geometrian lähtötiedoksi valittiin arkkitehdin pohjapiirustukset. Pohjapiirustuksille suoritetaan visuaalinen tarkastus, jossa tarkastetaan, että elementtien reunat ovat yksiselitteisesti havaittavissa. Suunnittelija määrittää itse elementtijaon, laattojen tukipinnat, tarvittavat elementtien välisten saumojen paksuudet sekä elementtien koot ja poikkileikkaukset. Tällä tavalla toimittaessa pystytään välttämään arkkitehdin mallintamisesta johtuvat virheet. Lisäksi jos geometrian lähtötiedot tehdään Teklassa mallintaen, vältetään ylimääräisen ohjelmiston käytöltä.

Ongelmana voidaan pitää pohjapiirustuksesta elementtien reunojen havaittavuutta. Jos reunat eivät ole yksiselitteisesti havaittavissa, tulee pohjapiirustus palauttaa arkkitehdille muutettavaksi. Pohjapiirustuksen eri viivojen merkitysten selvittämiseen voi kulua myös huomattavasti aikaa ja arkkitehti ei välttämättä ole tavoitettavissa juuri silloin, kuin tietoa tarvittaisiin. Toinen ongelma liittyy käytettyjen ohjelmistojen asetuksiin. Määritettäessä rakenteita viiva tai mallinnuspiste saattaa tarttua pohjapiirustuksessa väärään kohtaan. Pieniä piirto- tai mallinnusvirheitä voi olla hyvin hankala havaita etukäteen ennen algoritmin suorittamaa mallinnusta. Ratkaisu ongelmaan voisi olla ohjelmistojen kohtisuoruusasetusten (eng. ortho) hyödyntäminen. Lisäksi piirtäminen tai mallintaminen on tehtävä suurella huolellisuudella.

6.2.2 Algoritmiin ja mallintamiseen liittyvät ongelmat

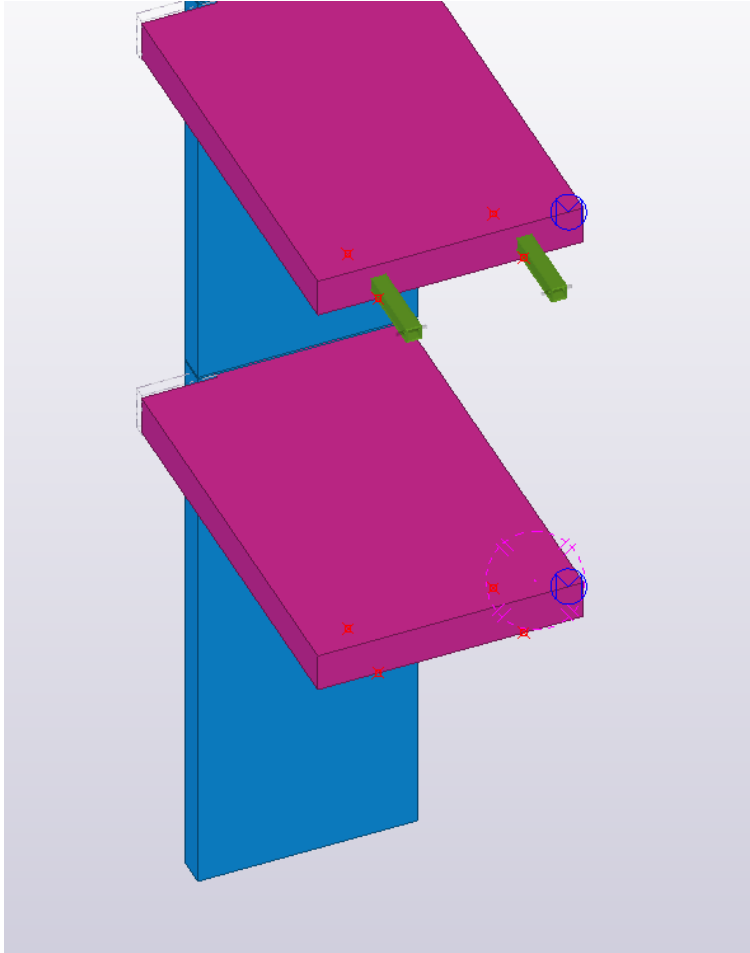
Grasshopper laskee algoritmin aina uudestaan, kun lähtöparametria muuttaa. Algoritmin koosta riippuen laskentaan voi kulua aikaa useita kymmeniä sekunteja. Jos muutettavia parametreja on useita, muutokseen voi kulua aikaa useita minuutteja. Aika ei välttämättä tunnu suurelta, mutta kun sitä vertaa yksittäisen komponentin suorittamiseen kuluvaan aikaan, joka yleensä vaihtelee millisekunnin ja sekunnin välillä, on se huomattava.

Erkkilä havaitsi tutkimuksessaan ongelman elementtijaon algoritmiavusteiseen suunnitteluun liittyen. Elementtijakoon vaikuttavat hyvin monet asiat (Erkkilä 2017, s. 67), joten sen algoritmin muodostaminen on erittäin hankalaa. Elementtijaon ongelma saadaan ratkaistua siten, että suunnittelija itse määrittelee elementtijaon. Toinen Erkkilän havaitsema ongelma on elementtien numerointi kerroksittain (Erkkilä 2017, s. 81-83). Samaa ongelmaa törmättiin myös tässä tutkimuksessa. Elementtien numerointiin kerroksittain

olisi mahdollista tehdä algoritmi, jolle määritetään ensimmäisen kerroksen elementtien start number ja start numberin askel. Ongelmaksi muodostuu algoritmin taipuisuus erilaisille kerrosten lukumäärille ja elementtien lukumäärälle kerroksittain. Algoritmia muodostettaessa olisi hyvä kiinnittää huomiota algoritmin taipuisuuteen, koska muuten algoritmia ei välttämättä voida hyödyntää tulevaisuudessa. Tästä syystä elementtien numeerointiasetukset jätettiin algoritmista pois.

Erkkilä esittää haastavana päättää mallinnetaanko kaikki kerrokset kerralla vai yksi kerros kerralla. Isoissa projekteissa suunnittelua tehdään lohko ja kerros kerrallaan, jolloin kerroksittainen mallintaminen olisi järkevää (Erkkilä 2017, s. 83). Tässä tutkimuksessa päädyttiin mallintamaan kaikki kerrokset kerralla. Parveketornit edellyttävät, että kaikki päällekkäiset elementit ovat samanlaisia, jolloin kaikkien elementtien yhtäaikainen mallintaminen on järkevää. Case-tutkimuksessa luotu algoritmi tehtiin sillä ajatuksella, että kaikki päällekkäiset elementit ovat samanlaisia.

Erkkilä mainitsee myös detaljointiin liittyviä ongelmia. Aukkojen leikkauksessa käytetty part cut -komponentti oli hidas ja jumiutui (Erkkilä 2017, s. 83). Tässä tutkimuksessa samaisiin ongelmiin ei törmätty lainkaan. Ongelmat luultavasti ovat ratkenneet Grasshopperin ja Teklan välisen linkin kehittymisellä. Erkkilällä oli myös ongelmia monitukaisten Tekla pluginien kanssa (Erkkilä 2017, s. 83). Tässä työssä käytetyt Tekla pluginit tarvitsevat joko pääosan ja kaksi pistettä tai pelkästään kaksi pistettä. Käytetyt pluginit saatiin toimimaan Grasshopperissa, mutta Teklassa ilmeni mallinnuksessa ongelmia. Parvekeputket mallintavassa pluginissa oli ongelmana, että kaikki mallinnetut putket eivät näy Teklassa. Mallinnuspisteet sekä pluginin symbolit näkyvät Teklassa, mutta itse putkea ei. Ongelma on esitetty kuvassa 48. Ylempään parvekelaattaan putket mallintuvat oikein, mutta alemmassa niitä ei näy. Ongelma toistui satunnaisesti eri parvekelaatoilla, kun algoritmilla mallinsi uudestaan. Satunnaisen esiintymisen seurauksena ongelman syiden selvittäminen on erittäin hankalaa eikä ongelmaan löydetty toimivaa ratkaisua tutkimuksen aikana.



Kuva 48. Parvekeputken mallinnuksessa ilmennyt ongelma.

6.2.3 Grasshopperin ja Teklan välisestä linkistä aiheutuvat ongelmat

Erkkilä mainitsi tutkimuksessaan Grasshopperin ominaisuudesta, joka suorittaa algoritmin uudelleen, kun Grasshopper avataan. Toinen Grasshopperin ominaisuus on, että se kirjoittaa Teklasta tehtyjen muutos yli, jos ohjelmien välistä linkkiä ei katkaista. Näistä voi aiheutua numeroinnin kanssa ongelmia, mikäli osasta elementtejä on jo tehty valmistuspiirustukset (Erkkilä 2017, s. 84). Tässä tutkimuksessa ei havaittu Erkkilän kuvaamaa ongelmaa Grasshopperia avatessa. Joka kerta kun Rhino ja Grasshopper avataan, pitää geometrian lähtötieto liittää Rhinoon uudestaan. DWG-tiedoston käytön yhteydessä ei havaittu ongelmia. Havaittiin kuitenkin ongelma käytettäessä Teklaan mallinnettua geometrian lähtötietoa. Kun Teklan ja Grasshopperin avasi uudestaan, algoritmi mallinsi parvekepielelementit, vaikka geometrian lähtötieto oli poistettu Grasshopperista ennen ohjelmien uudelleen avaamista. Ongelma esiintyi, kun käytettiin samaa Tekla mallia usean päivän ajan. Ongelma poistui, kun tehtiin uusi malli Teklaan.

Algoritmiavusteiseen elementtisuunnitteluun yleisesti liittyvä ongelma on muutokset elementeissä kuvatuotannon jälkeen. Elementit on suunniteltu algoritmiavusteisesti, valmistuskuvat on tehty ja pitäisi tehdä muutos. Silloin törmätään edellä mainittuun Grasshopperin ominaisuuteen, joka suorittaa koko algoritmin avatessa. Tällöin elementit mallintuvat uudestaan Teklaan. Tekla tulkitsee elementit muuttuneiksi, koska ne mallinnettiin uudestaan, jolloin numerointi menee sekaisin. Tällaisissa tapauksissa on syytä miettiä kannattaako muutos tehdä algoritmiavusteisesti vai perinteisesti mallintaa. Perinteisellä tavalla muutosten hallinta on huomattavasti helpompaa kuin algoritmiavusteisella menetelmällä.

Grasshopperin ja Teklan välisellä linkillä ei voida liittää valutarvikkeita elementteihin. Teklassa kaikki mallinnetut valutarvikkeet pitää liittää elementtiin, jotta ne näkyvät valmistuspiirustuksessa. Ongelma esiintyy parvekesaranoita mallintaessa. Kuten aiemmin case-tutkimuksessa kerrottiin, parvekesaranat mallinnetaan vain kahdella pisteellä custom componenttina. Koska ne mallinnetaan irrallisina custom componentteina, ne eivät liity parvekelaattaan. Parvekeputkien kanssa ei kyseiseen ongelmaan törmätty, koska putket mallintava plugin liittää ne automaattisesti parvekelaattaan. Tämä ongelma koskettaa kaikkia irrallisia custom componentteja. Custom componenttien liittäminen pääosaan pitää tehdä erikseen perinteisellä menetelmällä.

6.3 Johtopäätökset

Tässä kappaleessa verrataan case-tutkimuksen tuloksia case-tutkimuksen tavoitteisiin. Case-tutkimuksen tavoitteena oli kuvata parveke-elementtien algoritmiavusteinen suunnitteluprosessi, tutkia algoritmiavusteisen suunnittelun hyötyjä ja haittoja sekä tutkia käytettyihin ohjelmistoihin liittyviä rajoitteita. Verrataan myös case-tutkimuksen tuloksia kappaleessa 4.3 esitettyihin algoritmiavusteisella suunnittelulla tavoiteltaviin hyötyihin. Algoritmiavusteisella elementtisuunnitteluprosessilla pyrittiin nopeuttamaan ja tehostamaan raakamallinnusta sekä parantamaan raakamallinnuksen laatua.

Parveke-elementtien algoritmiavusteisen suunnitteluprosessin luomisessa hyödynnettiin Sweco Rakennetekniikka Oy:n elementtisuunnittelun prosessikaaviota. Prosessia testattiin case-tutkimuksen avulla. Suunnitteluprosessi ei merkittävästi muuttunut, mutta algoritmiavusteinen suunnitteluprosessi painottuu vahvasti raakamallinnukseen. Painotus oli tietoinen valinta, koska elementtien detaljointi on tehokasta jo nykyisen prosessin mukaan. Tutkimuksen lopputuloksena todetaan, että algoritmiavusteinen elementtisuunnitteluprosessi soveltuu hyvin parveke-elementtien suunnitteluun tiettyjen ehtojen täytyessä. Merkittävin ehto on päällekkäisten parvekkeiden vähäinen vaihtelu. Prosessi luotiin sillä ajatuksella, että päällekkäiset parvekkeet ovat samanlaisia.

Kappaleessa 4.3 ennakoitiin algoritmiavusteisella elementtisuunnitteluprosessilla saatavia hyötyjä. Case-tutkimuksen perusteella algoritmiavusteinen mallintaminen on tehokkaampaa, nopeampaa ja tarkempaa kuin perinteinen mallintaminen. Algoritmiavusteisesti voidaan vähentää inhimillisiä virheitä ja manuaalista työtä.

Työlle asetettiin teolliseksi tavoitteeksi luoda algoritmi parveke-elementtien mallintamiseen. Algoritmi luotiin parveke-elementtien suunnitteluun ja sitä ei voida suoranaisesti soveltaa muiden rakennusosien suunnitteluun. Kuitenkin se tarjoaa hyvät lähtökohdat jatkotutkimuksille ja uusien algoritmien luomiselle.

7. YHTEENVETO

Tutkimuksessa selvitettiin algoritmiavusteisen suunnittelun soveltuvuutta parveke-elementtien suunnitteluun. Aluksi tutkittiin algoritmiavusteisen suunnittelun taustaa ja teoriaa, nykyistä parveke-elementtien elementtisuunnitteluprosessia sekä parvekkeiden erityispiirteitä. Teoriaosuuden ja nykyisen elementtisuunnitteluprosessin perusteella luotiin parveke-elementtien algoritmiavusteinen suunnitteluprosessi. Case-tutkimuksessa tutkittiin algoritmiavusteisen elementtisuunnitteluprosessin toimivuutta. Case-tutkimuksen kohteena oli todellinen toteutunut asuinkerrostalo. Kohteesta saatiin arkkitehdin pohjapiirustukset sekä elementtien liitosdetaljit case-tutkimuksen lähtötiedoksi. Pohjapiirustuksen perusteella muodostettiin parveke-elementtien päägeometrialinjat. Ennen algoritmin luontia algoritmin haluttu tehtävä kirjoitettiin sanalliseen muotoon ja pilkottiin osakokonaisuuksiin. Osakokonaisuudet ratkaistiin erikseen ja niistä koostettiin monimuotoinen algoritmi. Case-tutkimuksen lopputuloksena todettiin, että kehitetty algoritmiavusteinen elementtisuunnitteluprosessi soveltuu hyvin parveke-elementtien suunnitteluun tietyin rajoituksin.

Tutkimuksen aikana ilmeni useita jatkotutkimusaiheita. Parvekelaattojen vedenpoiston kallistuksille ja kaadoille ei onnistuttu muodostaa algoritmia. Vedenpoiston algoritmiavusteinen mallintaminen voisi onnistua, jos siihen olisi sopiva Tekla plugin. Myös parveke-elementtien raudoitusten ja liitosten algoritmiavusteinen suunnittelu olisi toteutettavissa. Parveke-elementtien mitoitus on melko yksinkertaista, jos parvekelaatta on yksi aukkoinen. Parvekelaatan voimasuureiden laskenta voi vaatia FEM laskentaa, mikäli laatussa on pistemäisiä tukia, eli pilareita tai parvekeputkia. Myös koko parveketornin FEM laskennan voi toteuttaa algoritmiavusteisesti.

Case-tutkimuksen aikana heräsi ajatus, soveltuuko algoritmiavusteinen elementtisuunnitteluprosessi myös muille rakenneosille. Jatkotutkimuksena tulisi selvittää toistuuko tässä tutkimuksessa havaitut ongelmat myös muilla rakenneosilla tai ilmeneekö niillä uusia ennalta arvaamattomia ongelmia.

Algoritmin uudelleenkäytettävyys vaatii myös jatkotutkimusta. Esimerkiksi porrashuoneen ja hissikuilun algoritmiavusteinen elementtisuunnittelu olisi hyvä tutkimusaihe tässä tutkimuksessa luodun algoritmin uudelleenkäytölle. Porrashuoneessa ja parveketornissa esiintyy samankaltaisia piirteitä, koska molemmat koostuvat seinä- ja laattaelementeistä.

LÄHTEET

BETONITEOLLISUUS, R.Y., 2010. *Betonelementtiparvekkeet*.

ERKKILÄ, S., 2017. *Algoritmiavusteisen suunnittelun hyödyntäminen betonelementtirakenteiden suunnittelussa*, Tampereen teknillinen yliopisto.

HIRVIKOSKI, M., 2019. *Korkeiden rakennusten hankekehitysvaiheen stabiliteettimallin luonti algoritmiavusteisesti*, Tampereen yliopisto.

KARJALAINEN, P., 2018. *Kantavien rakenteiden algoritmiavusteisen rakennesuunnitteluprosessin kehittäminen*, Tampereen teknillinen yliopisto.

KETOLA, T., 2019. *Teräsristikon algoritmiavusteinen suunnittelu ja optimointi*, Tampereen yliopisto.

LALLA, A., 2017. *Kantavien rakenteiden parametrinen suunnittelu ja mallintaminen*, Tampereen teknillinen yliopisto.

MÄENPÄÄ, J., 2018. *Algorithm-Aided Structural Engineering of Steel-Framed Warehouse*, Tampereen teknillinen yliopisto.

ÖSTERLUND, T. and TANSKA, T., 2014. *ALGORITMIT PUURAKENTEISSA*.

PEIKKO FINLAND OY, 2016. *PS Parvekesarana, tekninen käyttöohje Versio: FI 9/2016*.

SWECO RAKENNETEKNIikka OY, *Sweco Rakennetekniikka Oy sisäinen materiaali*.

TOOLA, M., 2017. *Teräsrakenteisten siltojen parametrinen mallinnus*, Tampereen ammattikorkeakoulu.

TRIMBLE SOLUTIONS CORPORATION, , Grasshopper-Tekla Live Link. Available: https://teklastructures.support.tekla.com/not-version-specific/en/ext_grasshopperteklalink.

VÄHÄNEN, P., 2019. *Parametrisen suunnittelun hyödyntäminen teräsbetonisten runkorakenteiden luonnossuunnittelussa*, Oulun yliopisto.